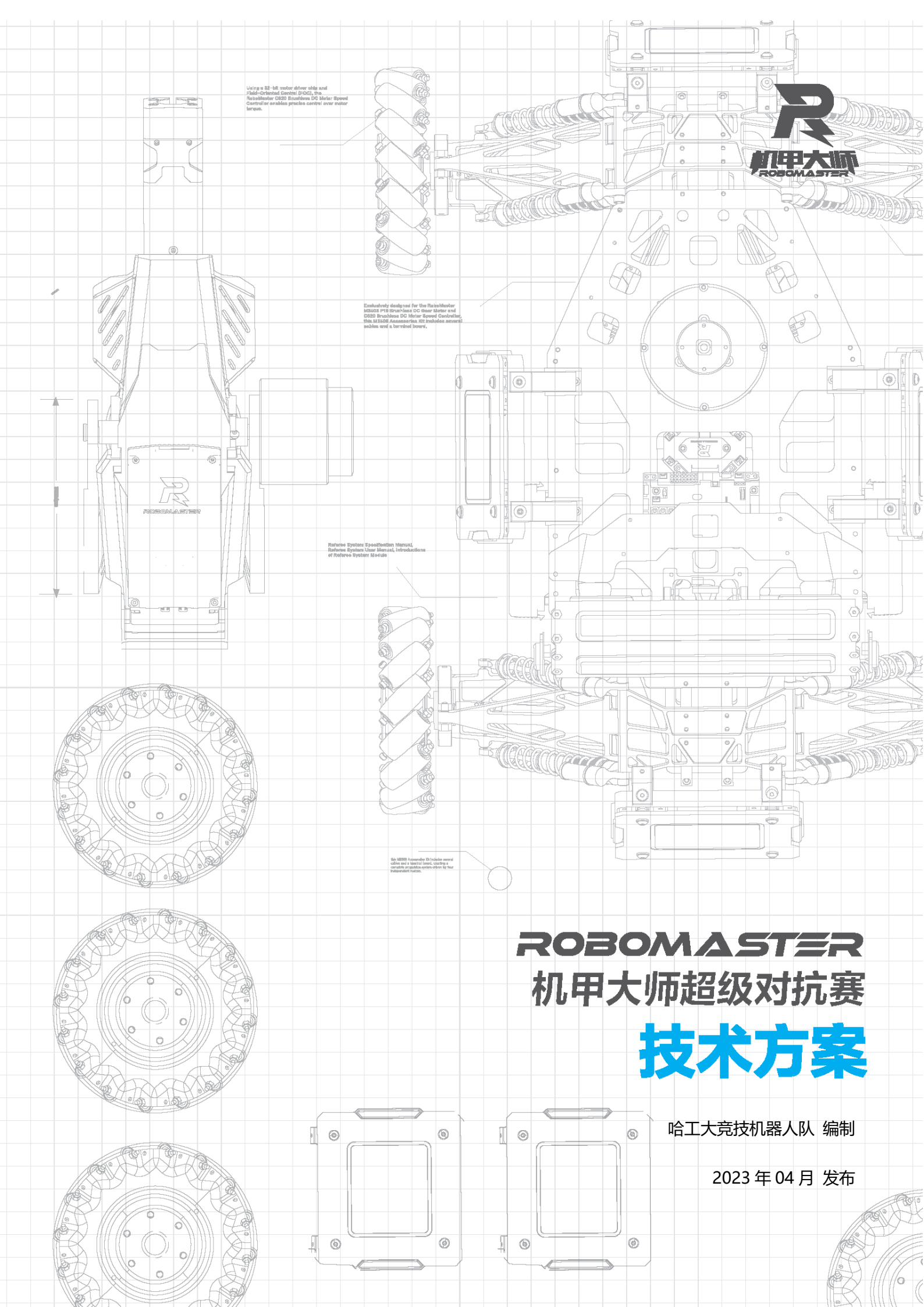


ROBOMASTER
机甲大师超级对抗赛
技术方案

哈工大竞技机器人队 编制

2023 年 04 月 发布

前言

本技术方案报告由哈尔滨工业大学 I Hiter 编制，适用于 RoboMaster 2023 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	撰写人员 3
机械	刘峰荣	姚宇轩	周柏宇
硬件	李浩田	单俊奇	
软件	张睿成	王阳	
感知（车载端）	牛傲宇	张浩	龚林奥
感知（神经网络）	李乐乐	韩镐轩	陈熠琳
定位	周兰欣		
规划	王阳		
决策	徐朋	符兴	
其他部分	周柏宇	张天奕	王燚辰

目录

1. 概述

1.1 背景&目标

当前赛季的哨兵机器人是全兵种中改动最大的、涉及功能最多、最自动化、最智能化的兵种。根据超级对抗赛规则手册 1.2[1]，哨兵机器人在己方前哨站被击毁前，处于无敌状态。因此，在战术中的定位为对敌方机器人造成伤害乃至持续高强度压制的主要输出兵种，同时肩负补足建筑物伤害、守护我方前哨站的辅助功能。

由此引申出本赛季哨兵机器人的两个技术目标，分别为底层结构的稳定性、高上限和上层控制策略的全面性、自动化。底层结构包含全车的机械部分、软硬件控制部分。这部分决定了整车的运动性能以及可实现的战术目标的上下限。上层控制策略包含算法层的感知、定位、规划、决策四大块。这部分决定了整车实际比赛中实现的战术策略及面对复杂情况的灵活性。

1.2 其它学校机器人分析综述

由于哨兵机器人涉及技术栈较多，下面简略分为机械、电控软硬件、视觉、导航及决策四方面进行比较和叙述。

机械方面，由于哨兵改动较大，主要参考方向为其他学校哨兵或步兵的供弹形式、其他学校普通步兵的底盘结构、双枪步兵的切枪机构、其他学校自动步兵的传感器布置等。首先供弹方式可分为三种，例如常见步兵的上供弹形式（图 1.2.1 左）以及哨兵半下供弹形式（图 1.2.1 右），这两者在使用时，存弹量波动都会对 Yaw 轴、Pitch 轴电机产生负载变化。对于本赛季需要预装填 500~1000 发弹丸的哨兵机器人来说，并不是非常优秀的设计方案，因此不采纳。

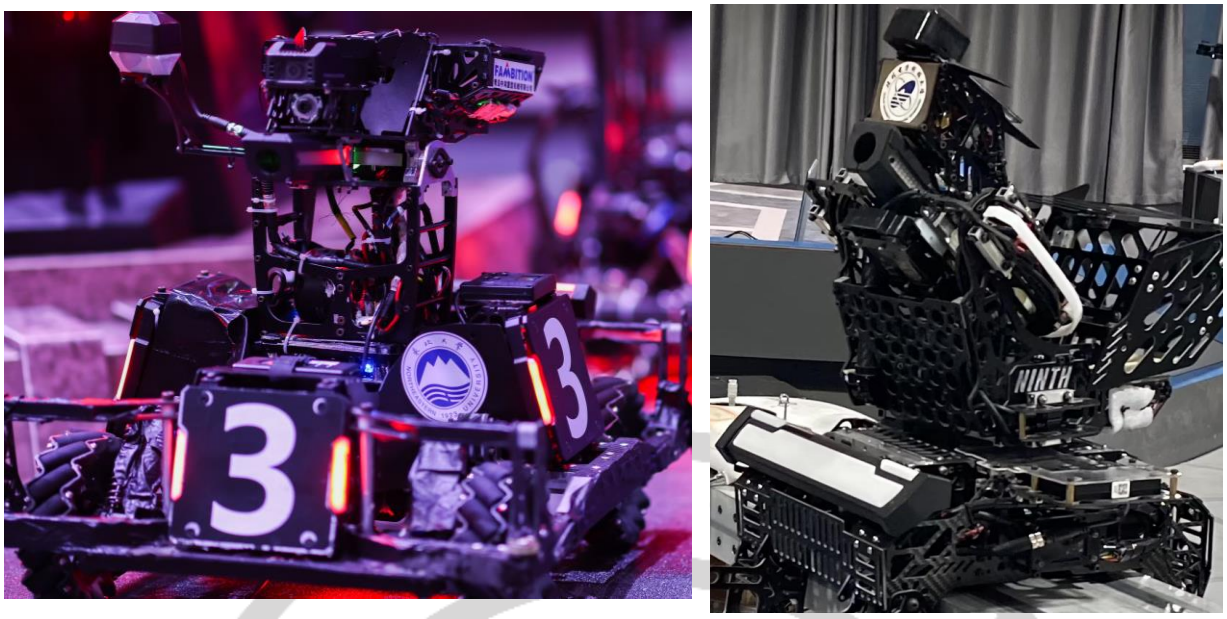


图 1.2.1　左：东北大学 2022 赛季步兵机器人 上供弹云台图
右：桂林电子科技大学 2021 赛季开源哨兵机器人[2] 半下供弹上云台整体图

又有部分学校哨兵采用过下供弹方案，例如，浙江大学 Hello World 战队在 2022 年联盟赛技术答辩中的哨兵机器人上云台（图 1.2.2），采用碳板支架大弧度链路的形式实现了小弹丸下供弹的效果；同时链路在俯仰角过渡处采用上下双动无联动的弧线形碳板（图中红框处）。大弧度链路部分值得采纳，但俯仰过渡形式并不匹配于新哨兵。究其原因为：旧哨兵机器人处于轨道上方较高处，工作俯仰角为-10°及以下；但新哨兵机器人工作俯仰角应与步兵机器人相近，大约为±30°附近，显然上图中弹丸由于始终可以依靠自重保证下表面与下过渡板的贴合，从而即便限位较为宽松的情况下依然可以实现较好的导向效果。但若俯仰角改为较大范围，需要考虑在最大仰角处最前弹丸的受力主要是后方弹丸斜向上的推力和上过渡板斜向下的推力，限位过宽的情况可能导致弹丸在该拐角处堆挤，形成卡弹。

图 1.2.2　浙江大学 HelloWorld 战队 2022 开源哨兵机器人[3] 上云台供弹链路部分细节

底盘结构方面，根据轮系种类可以简单划分为麦克纳姆轮底盘、全向轮底盘、舵轮底盘。由于控制困难而且没有特殊加成，这里忽略平衡类底盘（二轮结构、共轴麦轮结构等）。相同点为他们都可以实现真正意义上的“全向运动”，麦克纳姆轮通过四个轮子之间速度抵消与合成来实现空间内任意方向的目标速度矢量；全向轮类似，只不过单个辊子的受地面摩擦力方向与轮缘线速度方向夹角为0°。而舵轮每个轮系存在两个自由度：舵向与航向，舵向只负责调控轮的转速，航向只负责调控轮的朝向。由于仅是简单方案比较，下面简单叙述此三种轮系底盘的逆运动学解算。首先定义车体坐标系、各轮子速度v_{fl}、v_{fr}、v_{rl}、v_{rr}、v_1、v_2、v_3、v_4如下图所示：

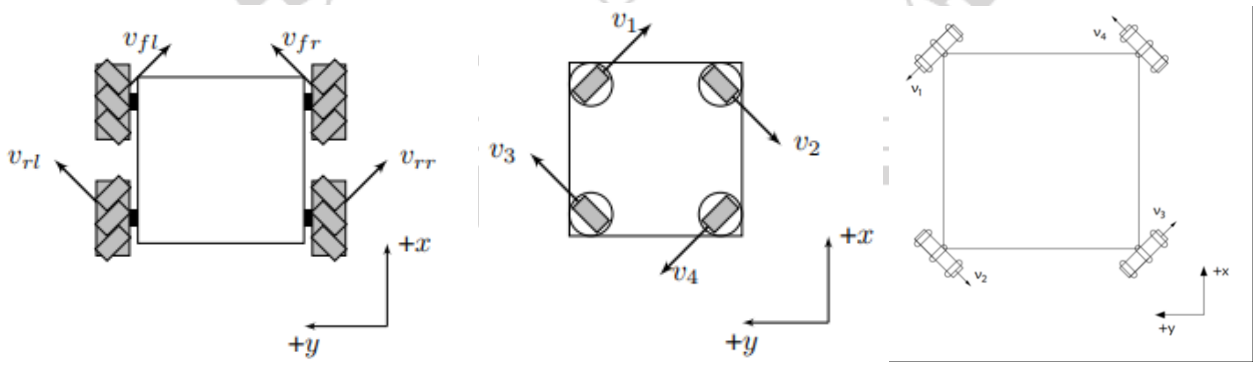


图 1.2.3　左：麦克纳姆轮底盘受力分析图
中：舵轮底盘受力分析图
右：全向轮底盘（四轮）受力分析图

经过简单的物理推导，麦克纳姆轮底盘的逆运动学解算[4]为：

$$\begin{bmatrix} v_{fl} \\ v_{fr} \\ v_{rl} \\ v_{rr} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & -(r_{fl_x} + r_{fl_y}) \\ 1 & 1 & (r_{fl_x} - r_{fl_y}) \\ 1 & 1 & (r_{fl_x} - r_{fl_y}) \\ 1 & -1 & (r_{fl_x} + r_{fl_y}) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

同理，舵轮底盘的逆运动学解算[4]为：

$$\begin{bmatrix} v_{1_x} \\ v_{1_y} \\ v_{2_x} \\ v_{2_y} \\ v_{3_x} \\ v_{3_y} \\ v_{4_x} \\ v_{4_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -r_{1_y} \\ 0 & 1 & r_{1_x} \\ 1 & 0 & -r_{2_y} \\ 0 & 1 & r_{2_x} \\ 1 & 0 & -r_{3_y} \\ 0 & 1 & r_{3_x} \\ 1 & 0 & -r_{4_y} \\ 0 & 1 & r_{4_x} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

同理，全向轮底盘的逆运动学解算[5]为：

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & r \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & r \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & r \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & r \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

假定三种底盘的r是相等的，给定一组相同的目标速度，比较三种底盘电机需要输出的速度。显然，舵轮需要输出的速度是最低的，即舵轮在直线行驶方面最省功率。因此机械方面为了实现哨兵机器人的高上限，选用舵轮底盘。

另外一点十分重要的是，舵轮结构为为数不多可以实现瞬间急刹车的底盘形式，这点会在路径规划算法及近距离避障中发挥很重要的部分。在 RoboMaster 的历史中，舵轮步兵出现的场次并不多，其中结合各队伍展示的舵轮车辆，主要区别点在于悬挂和舵向动力布置形式（图 1.2.4）。例如，伊利诺伊香槟大学首次在正赛中使用非 6020 作为舵向动力源[6]，这会引入一个舵向上电位置标定的问题，他们的解决方案是增加一个绝对值编码器（图 1.2.4 左 绿色部分）但这样会引入硬件系统的不稳定。华南理工大学的舵轮步兵的动力单元整体布置结构繁杂，主要是由于增加了双摇臂舵机救援模块[7]（图 1.2.4 中 轮上方）。这两套舵轮悬挂都有一个共同弊端，即无法保证轮回转轴在悬挂压缩的任意行程都保持距离云台回转中心不变。引入的风险便是在起伏路段中小陀螺、亦或是部分悬挂出现了故障的情况下，无法正常

完成小陀螺的同时击打的过程。这也许在有操作手的步兵中可以接受，但在全自动的哨兵中是需要尽量避免的存在。而在重庆大学的悬挂形式[8]（图 1.2.4 右 轮侧），采用了内燃机车中常用的一种连杆约束——瓦特连杆，可以实现短距离内悬挂行程的类竖直约束，这值得我们学习。

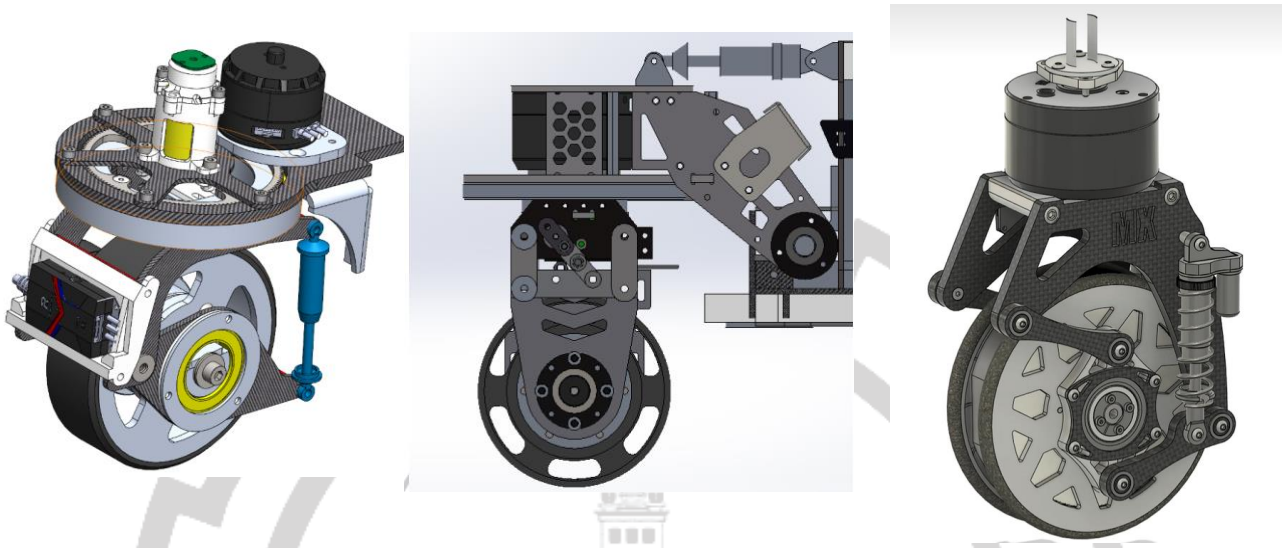


图 1.2.4　左：伊利诺伊香槟大学 2022 开源 非 6020 舵轮结构
中：华南理工大学 2021 开源 国赛舵轮步兵 动力单元部分
右：重庆大学 2022 视频展示 差速舵轮概念模型

本赛季哨兵机器人共需要安装两个机动枪口，根据战术需要布置为单云台双发射机构的形式。因此会涉及到一个双发射机构的切换模块，这个模块要求了高重复定位精度以及较高的机械可靠性。目前双发射机构主要是两个方案，以深圳大学、东北大学为主的双枪管不切换方案；以哈尔滨工程大学为主的单枪管切换方案。后者的好处在于弹道散布是前者的一半，同时过高的击打频率会导致装甲板不识别。

图 1.2.5　左：深圳大学 2021 开源双枪步兵[9] 实物图
右：哈尔滨工程大学 2022 开源双枪平衡步兵[10] 云台部分模型图

电控软件方面，哨兵机器人的改动主要针对底盘方面的控制策略。因此电控侧今年并无许多参考资料，相关可参考部分为论坛的舵轮底盘控制算法开源，例如浙江大学 Hello World 战队在线文档[11]中有对舵轮底盘从逆运动学解算到运动优化算法较为细致的说明。华南理工大学[12]、东北大学[13]等都对舵轮底盘的控制算法有过开源。

电控硬件方面，队内的控制板自主程度很高，并且经过了多达三个赛季的稳定使用，因此无需参考任何其他学校的控制板方案。超级电容方面同理，但由于激光雷达为精密传感器，过高的速度会带来精度较大的损失，现阶段无需超级电容便已经满足了哨兵的功率需求。在传感器上，多数学校采用了近期推出的 Livox Mid-360 激光雷达作为车辆的主定位、建图传感器（图 1.2.6 左）。也可以看到一些此前做过自动步兵方案的学校使用过其他型号的测距传感器，例如浙江大学 Hello World 战队[14]使用乐动 LD06 激光雷达与 Intel RealSense T265 深度相机作为定位及建图的设备（图 1.2.6 右）。但这两者点云密度都不够高，在后续提速过程中，可能会产生定位误差大甚至漂移严重的问题，无法满足使用需求。

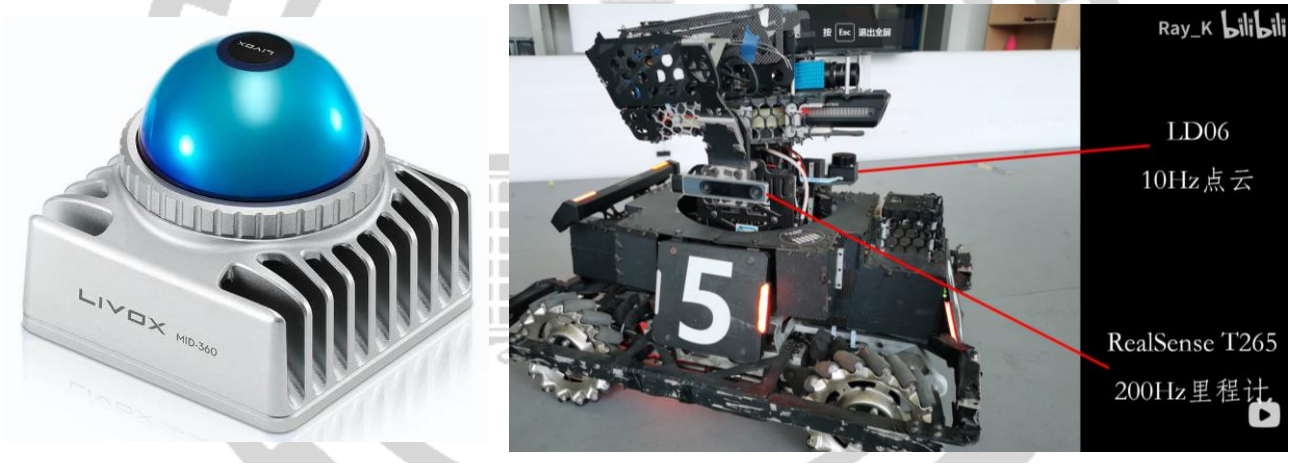


图 1.2.6　左：Livox Mid-360 激光雷达 实物图
右：浙江大学 2022 自动步兵展示视频 传感器布置图

视觉感知方面，传统视觉上还是使用了沿用已久的 cv.solvePnP 函数实现单目相机的位姿解算，未对其他学校方案进行过多分析。数字识别中使用的是基于 Google 提出的 MobileNet 轻量级神经网络进行训练，并使用 OpenVINO 部署，对模型进行修改后用低曝光的数据集进行训练。

导航及决策方面，在定位建图中，我们主要参考了两个开源算法，香港大学的 r3live[15]和 Fast-lio 算法[16]。最初使用 Livox Horizon 作为建图设备，发现 r3live 算法在同时旋转和平移即较大幅度转弯的时候会让里程计直接崩溃。在路径规划中，全局路径规划比较了三个算法：A*算法[17]、RRT 算法[18]和蚁群算法[19]，其中 RRT 算法的本质是随机扩张搜索路径，因

此很多改进后的算法依然存在收敛时间较长的缺陷；而蚁群算法的计算量较大，不适于在这种算力受限的自主移动小型机器人上使用。局部路径规划中，同样比较了三种算法：动态窗口法[20]、遗传算法[21]、人工势场法[22]。单一的路径规划算法都存在其局限性，为了进一步提高路径规划算法的整体性能，混合路径规划算法近年来成为热点研究方向，研究人员将许多不同的算法进行组合，均得到了很好的路径规划效果。例如中国农业大学的劳彩莲提出了一种基于 A*算法与动态窗口算法的混合路径规划算法[23]，所规划路径相对于传统 A*算法更为平滑与高效。在决策中，主要参考了哈工深 RMUA 队伍的开源代码[24]，即基于行为树搭建的决策模块。

1.3 机器人功能定义

哨兵机器人功能定义、设计需求见下图。

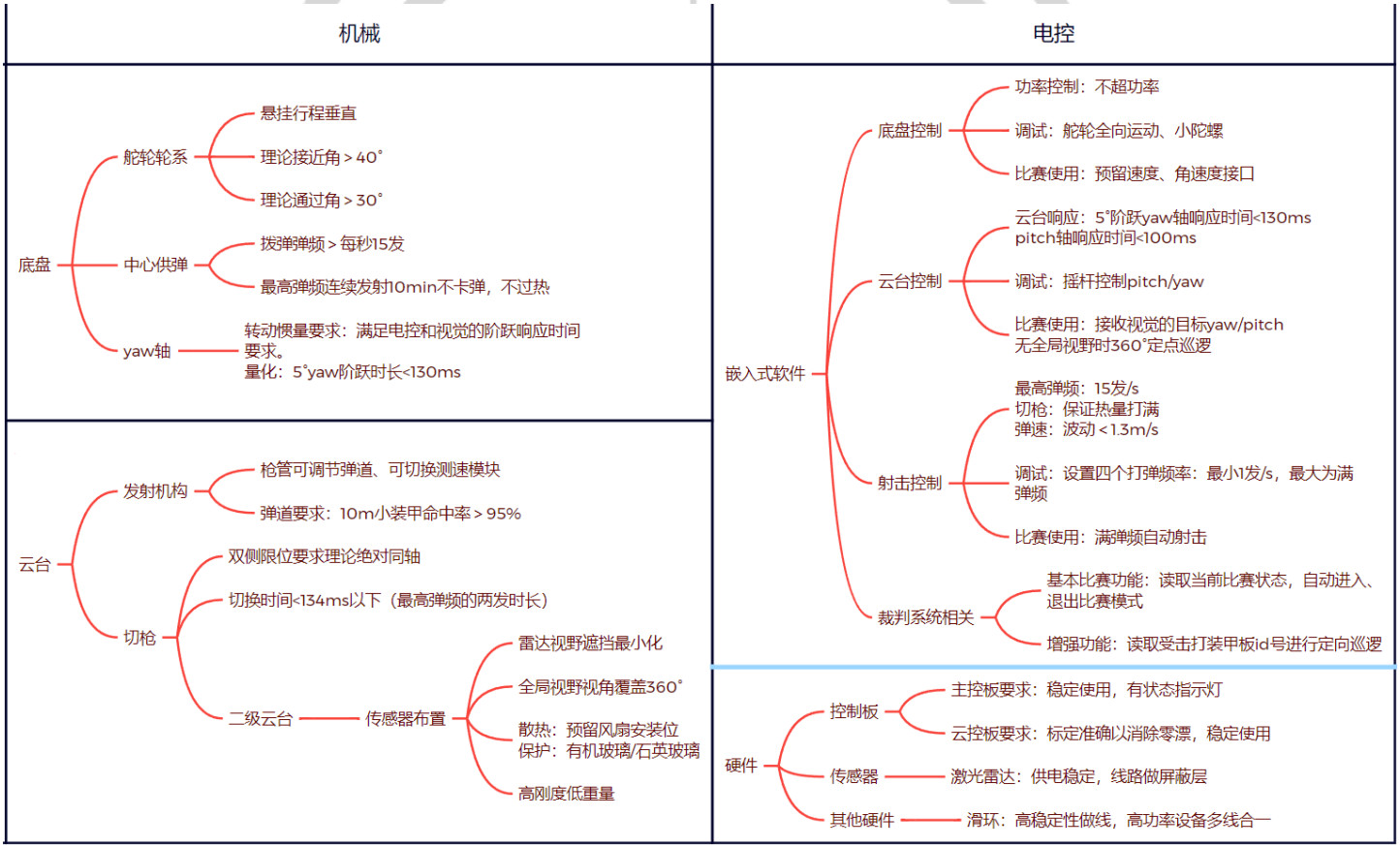


图 1.3.1　哈尔滨工业大学 2023 哨兵设计需求

1.4 机器人核心参数

表 1.4.1 哈尔滨工业大学 2023 哨兵机器人 核心参数表

名称			参数
重量			$20.7kg$
重心			重心高度$230.45mm$
尺寸（长*宽*高）			$667*610.7*695mm$
俯仰角范围			$-29°\sim+34°$
接近角			$31.27°$
通过角			$22.26°$
电机用途及数量	**底盘电机（数量）**		M3508（8 个）
	拨弹电机（数量）		M2006（1 个）
	云台电机（数量）		GM6020（3 个）
	发射电机（数量）		M2006（2 个）
车载传感器	**激光雷达**	OUSTER OS0	垂直视野 $90°$ 水平视野$360°$ 范围$0.5m\sim35m$ 精度$\pm0.1cm$
辅瞄相机	**工业相机**	大华	焦距$8mm$ 抓图帧率$208fps$
全局视野相机	**工业相机**	迈德威视	$Fov120°$ 抓图帧率$30fps$
	工业相机	迈德威视	$Fov120°$ 抓图帧率$30fps$
	USB 相机		$Fov90°$ 抓图帧率$30fps$

1.5 设计方案

1.5.1 机械结构设计

根据前文设计需求，现展示哈尔滨工业大学 2023 哨兵一代车（图 1.5.1）。

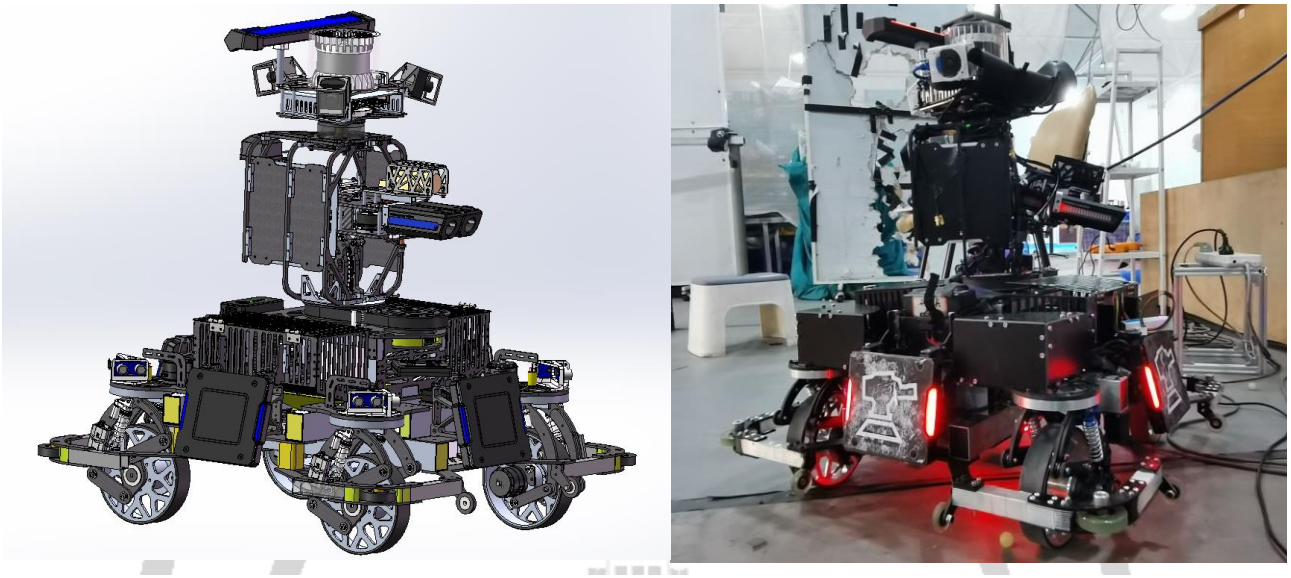


图 1.5.1.1　左：哈尔滨工业大学 2023 哨兵一代车 模型图
右：哈尔滨工业大学 2023 哨兵一代车 实物图

1.5.1.1 供弹部分设计

本赛季哨兵兵种取消了哨兵轨道，改为了在地面上进行自主移动。由于其初始允许发弹量为 750 发，需要预装填的弹数很多，若采取上供弹方式或半上供弹方式，会占用云台较大的空间，并且使整车的质心偏高，增大云台的转动惯量，降低云台转动和底盘小陀螺的稳定性，于是本战队采用小弹丸中心供弹的方式。这种方式可以降低整车的质心，并使质心几乎位于整车底盘的几何中心处。并且，中心供弹方式将云台的弹仓转移到了底盘，为全局视野相机和激光雷达等在云台上的安装，提供了空间，缩小了哨兵的体积，相比于其它供弹方式有很大的优势。

中心供弹原理

中心供弹拨盘主要由拨齿和轨道组成（图 1.5.1.2）。轨道 1 是弹丸运动的主平面，轨道 2 是弹丸进入轨道 3 前经过的90°水平面圆弧轨道，轨道 3 是最后的90°竖直平面上升弯道。小弹丸沿着轨道 1 被拨齿逆时针从四周向中心推动，在轨道 2 时经由后面的弹丸推动向中心旋进，将前面的弹丸推动沿轨道 3 继续向上，实现中心供弹。

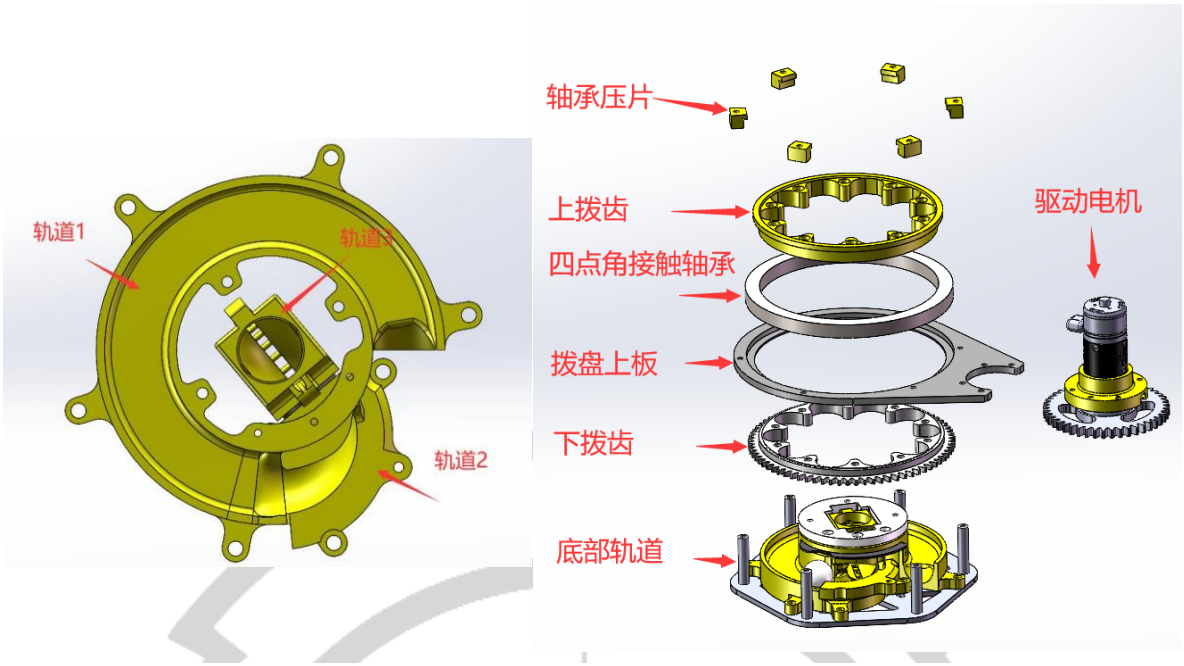


图 1.5.1.2　左：中心供弹轨道模块组成 模型图
右：中心供弹拨盘总成 爆炸图

拨盘主要由底板和拨盘上板进行整体的固定，上拨齿和下拨齿将轴承上下夹紧，轴承又通过弹仓压紧在上底板上，从而实现整体的固定。同时下拨齿内侧是拨齿，外侧与电机是相啮合的一组齿轮，实现传动。在底盘底部只需拧下与铝柱相连接的螺栓，即可取下驱动电机、底部轨道、底板的整装模块，实现快拆。

拨盘采用双层设计，可缓解弹频过高时的空拨问题。轨道 1 首端设有退弹槽，当拨盘中有异物导致卡弹时，电机反转，拨齿将弹丸向后拨，退出弹丸，排出异物。在一定程度上可

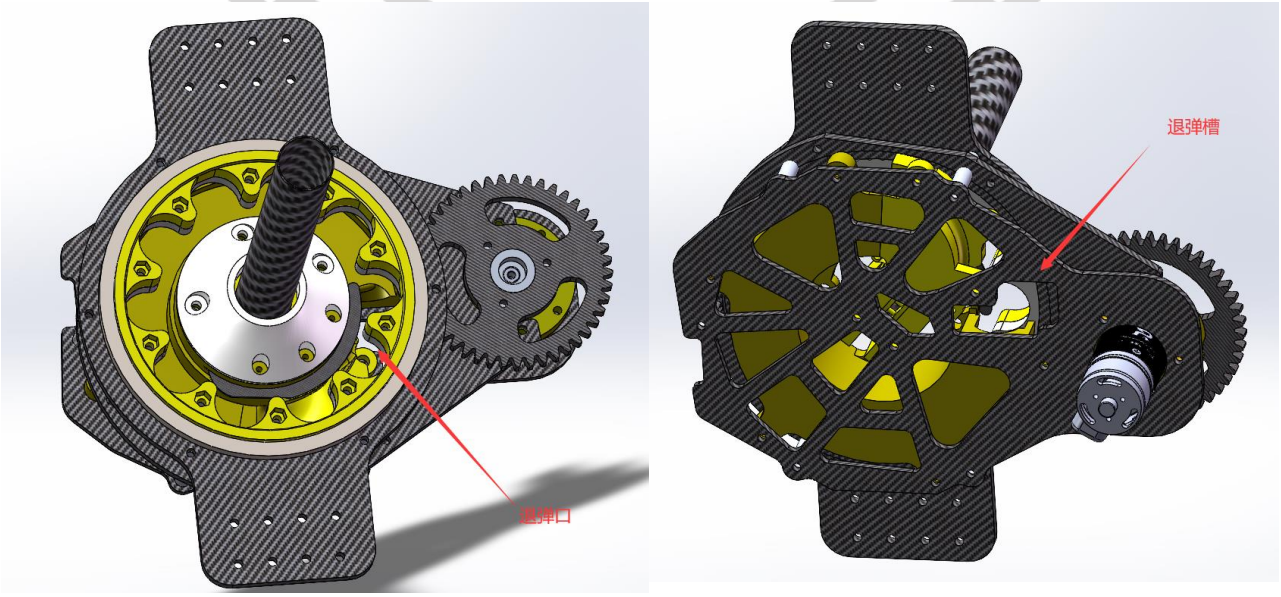


图 1.5.1.3　左：中心供弹总成 轨道退弹口示意
右：中心供弹总成 底板退弹槽示意

解决异物卡弹的问题，避免拨盘卡死后只能手动拆卸拨盘。轨道 2 的上方置有碳片，辅助分层，避免二层弹丸下落与轨道 2 卡死。

测试情况

在拨盘测试过程中，卡弹的原因主要有在分层开始处弹丸与碳片拨齿之间卡死、跳弹卡死两点。

弹丸与链路 2 上方接触点过高容易跳弹，在适合的位置用打印件覆盖对弹丸运动进行了限制，彻底解决问题。另外，在弹丸开始分层处，由于上方小弹丸掉落速度有限，弹频较高，在掉落过程中与分层片边缘接触并且与拨齿内壁接触，二力通过球心达到平衡而卡死，可以通过加大分层片挠度解决，例如给碳片开释放槽、更换弹簧钢材质等。

改进后的拨盘在测试中，在 13 发/s 的弹频下，可以连续打弹十分钟而不卡弹，即 11800 发左右，满足了设计需求。

1.5.1.2 底盘部分设计

底盘设计分析

底盘是哨兵机器人的移动机构，承担承载哨兵机器人的其他机构进行平移、行驶盲道、飞坡、场地交互、保护整车等任务。基于以上功能需求，对底盘设计进行分析。

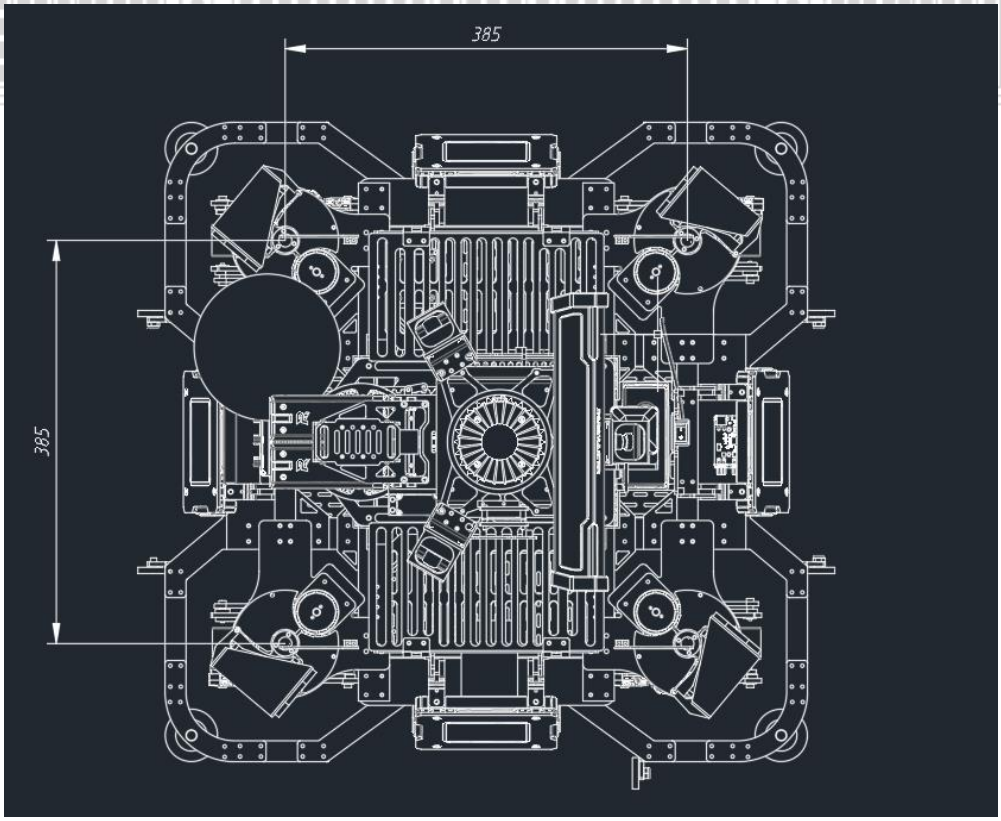


图 1.5.1.4　哨兵机器人 二维俯视图

铝架设计

底盘铝架主要的载荷为各种竖直方向的冲击载荷和垂直于铝管截面方向的冲击载荷。由于铝方管的抗失稳能力很强，因此常见的失效形式为竖直方向的铝管弯曲变形。

根据简单的材料力学可知，抗弯性能主要取决于抗弯刚度EI，其中E为弹性模量，与所选材料直接关联；I为材料极惯性矩，由于受载主要为下图y方向，故变形方向的极惯性矩为

$$I_z = \frac{BH^3}{12} - \frac{bh^3}{12}$$

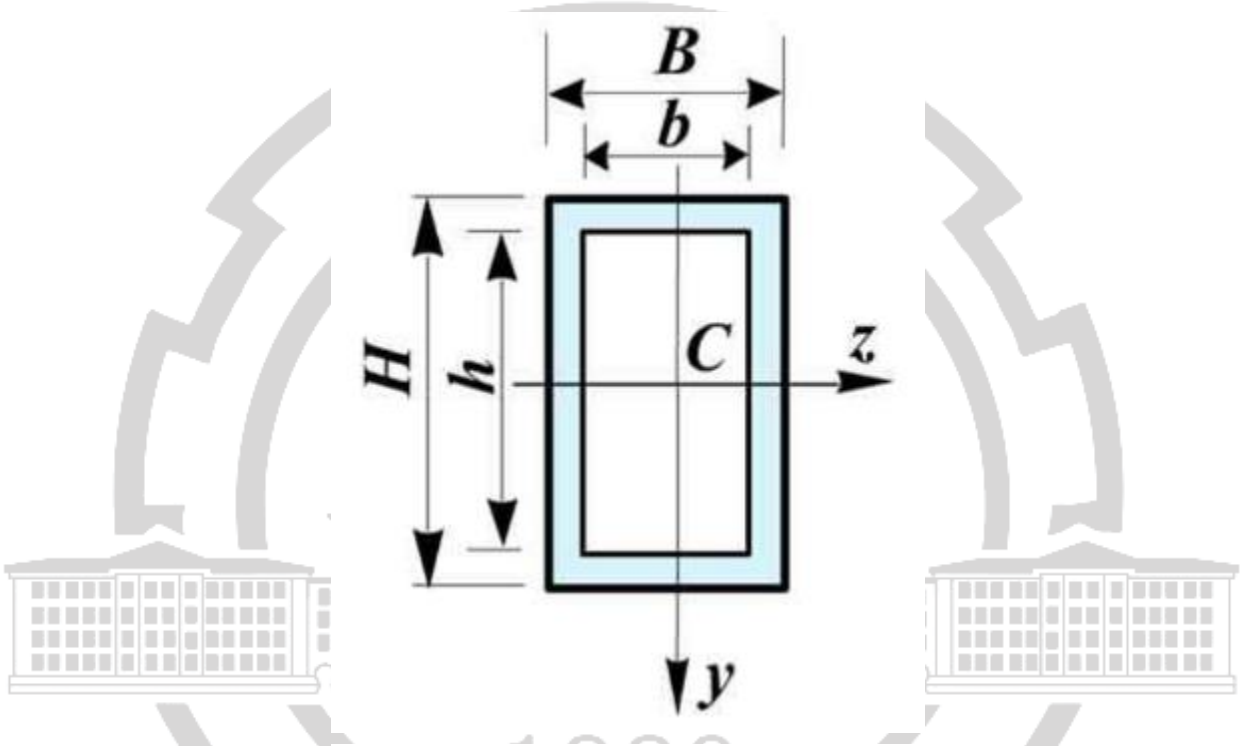


图 1.5.1.5　空心矩形截面管的形状参数

为了增强铝架的强度和刚度，哨兵机器人的底盘铝架使用“井字形”铝管搭接，铝管使用$30 \times 40 \times 1$的铝方管，相比传统的$20 \times 20 \times 1$的铝管，理论上抗弯刚度提升到原来的6.97倍。井字形也方便舵轮正方形的轮距分布和中心供弹的布置。同时，在固定轮系的铝管中添加了 PLA 打印件，解决铝管压溃的问题。在铝管的连接方式上，使用环氧树脂 AB 胶进行胶接，搭配抽芯铆钉铆接的形式，实现了轻量化高强度的效果。铆接碳板在铝管所在位置铣存胶槽（图 1.5.6 右 红框）、打定位孔定位，提高安装精度，与焊接相比，这种搭接方式抗弯强度更高，减小了铝管受力弯曲的程度，同时避免铝管在焊接过程中受热的问题。

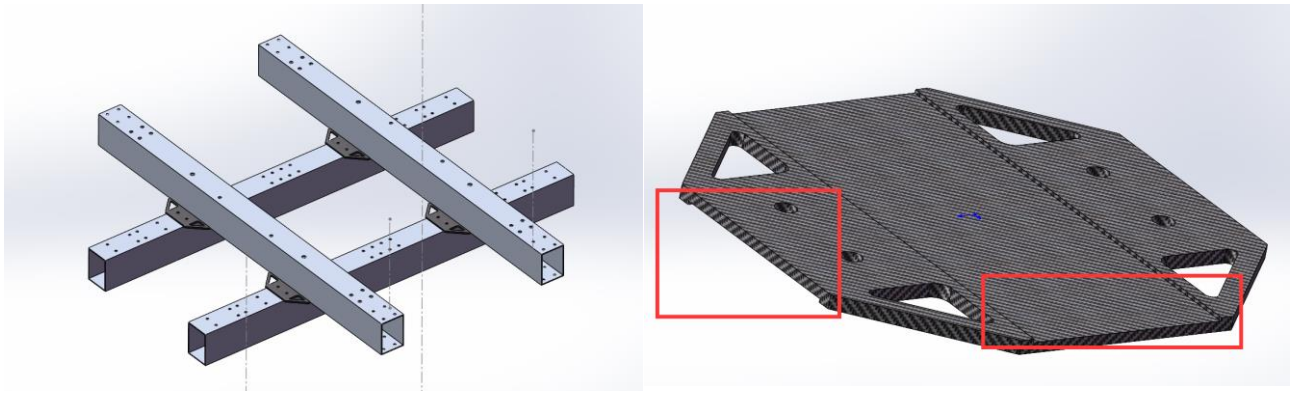


图 1.5.1.6　左：哨兵机器人底盘铝架 模型图
右：铆接碳板 模型图

为了更好地保护整车，在防撞管的选择上，对比常规铝管，我们选取了高塑性、高刚度、质量轻的 PC 管。表面粘接横纵双层纤维胶带，进一步补足该 PC 管的韧性。

在导轮设计上，我们在车身前后左右四个方位各放置了两个导轮，在上坡度较大的坡时，由于导轮的作用，增大了其通过角，辅助哨兵机器人更好地上下坡。同时，这些导轮也有助于飞坡，当飞坡姿势不佳时，导轮触地，通过导轮的作用，可以使哨兵机器人平稳落地。其次，在机器人从台阶掉落时，导轮也可以起到保护作用，有很强的实用性。另外，在四个角的方位上，我们横向放置了大导轮，以便于哨兵机器人发生碰撞时起到保护作用，避免小陀螺及大角度转弯时被卡住。

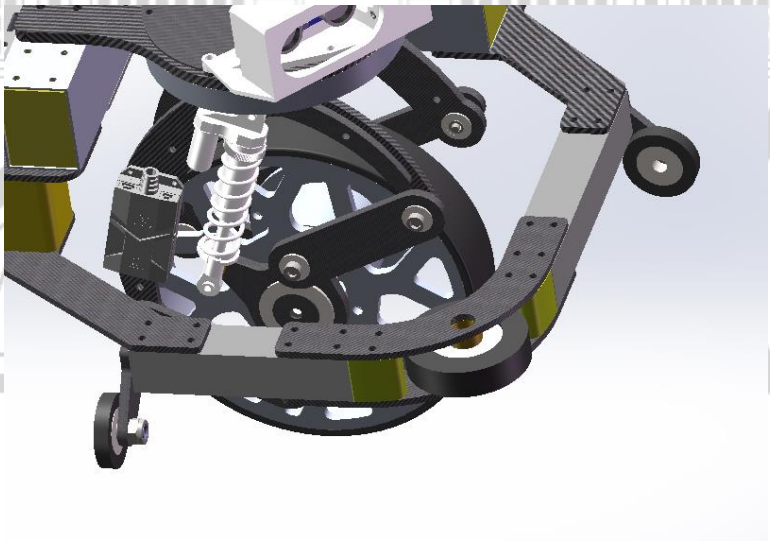


图 1.5.1.7　哨兵机器人底盘导轮 局部图

轮系设计

1.舵轮轮系设计

底盘最主要的设计部分在于一个好的轮系，它决定着机器人移动的机动性与稳定性。针

对哨兵机器人的底盘需求和功率，选择使用舵轮轮系。相比于麦轮，舵轮没有速度损耗，麦轮由于辊子的倾斜而产生速度损耗，能充分利用底盘功率。航向电机选择 3508 电机直连，3508 扭矩与转速足够，有较好的稳定性。航向电机选择 3508 电机，齿轮传动。相比 6020 电机质量更小，但每次上电需要进行标定，且齿轮设计上难于保证与轴承同轴心。

轮毂直径参考了 21 年的舵轮步兵的轮毂直径，舵步的舵轮直径为$120mm$,底盘大小为$532mm \times 526mm$,哨兵底盘大小为$610mm \times 615mm$，所以选择舵轮直径为$140mm$。

舵轮包胶方面，由于硬度越大其峰值速度越快，但抓地性能变差，加速能力降低，所以选择使用肖氏硬度 90A，并进行表面磨砂处理。在拥有足够硬度的同时，增加抓地力。

为方便航向 3508 电机和电调的走线，在转轴中心放置小滑环，使用六面螺母，垂直放置 L 形走线板，打出走线孔，便于滑环走线。

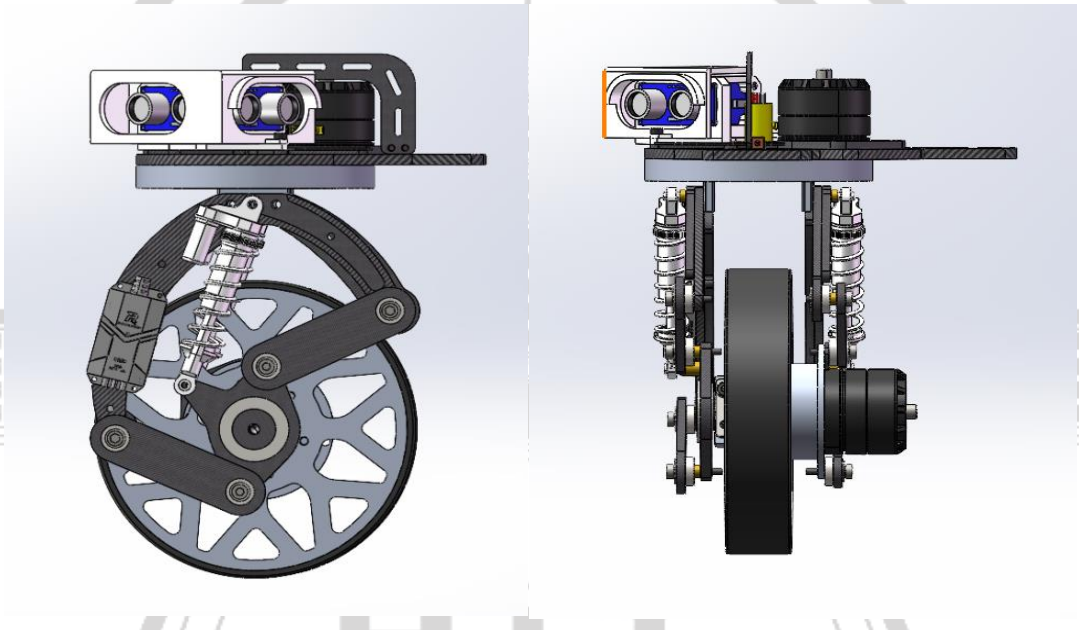


图 1.5.1.8　哨兵机器人底盘轮系模块 模型图

2.瓦特连杆悬挂设计

本舵轮轮系的主要特点在于在悬挂方面采取了双端特连杆机构，两侧都安有瓦特连杆和避震器，若使用单端支撑的方式，轮子收到地面向上支持力的作用，避震器压缩在轮单侧施加向下的力，两个力形成一个力偶，施加在悬挂板上造成形变；而双端支撑的结构在轮子两侧都形成一个向下的力，两个力对称，避免产成力偶，这样就优化了受力结构，不会有单端支撑的“内外八”现象，同时瓦特连杆机构保证了轮系只能在垂直地面方向上下移动，使各个轮距始终保持不变，四个轮距形成正方形，有利于小陀螺的稳定性和电控的底盘解算。

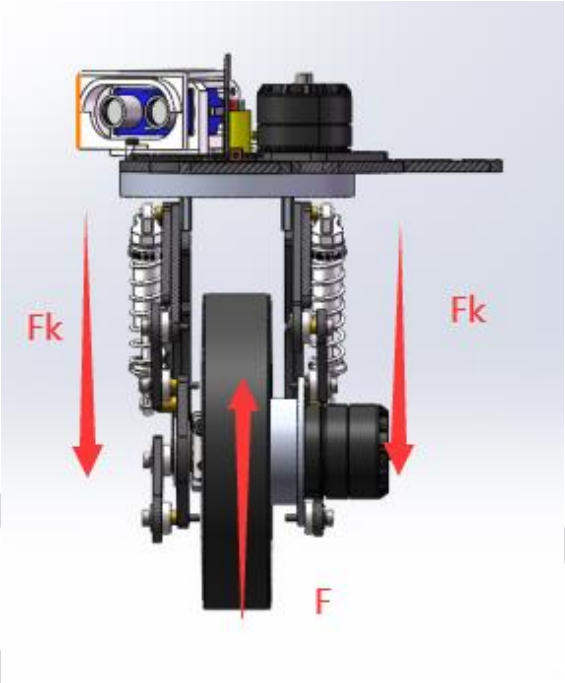


图 1.5.1.9　哨兵机器人轮系受力分析

瓦特连杆是一种平面四连杆机构，在运动过程中中间连杆中心点轨迹形成“类 8 字形”（图 1.5.1.10）。

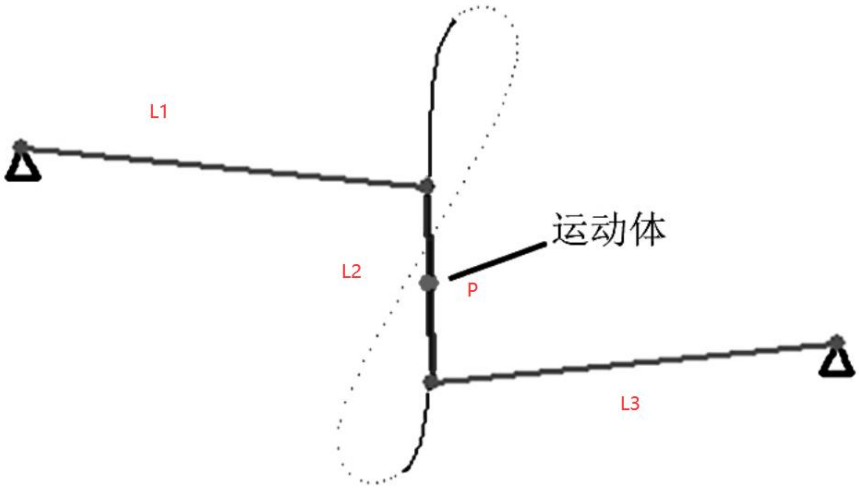


图 1.5.1.10　瓦特连杆机构中心点P运动轨迹示意图

其中有两段类直线行程，我们取其中垂直运动的部分。因此只要在设计时将轮心位于P点，配合悬挂行程，即可保证轮子只进行近似于垂直的直线运动，其中$L1 = L3$。当$L1$与$L3$不相等时，也有两段直线行程，但P点将不在$L2$的中心处，而是靠近杆更长的一端，这样不方便进行设计。

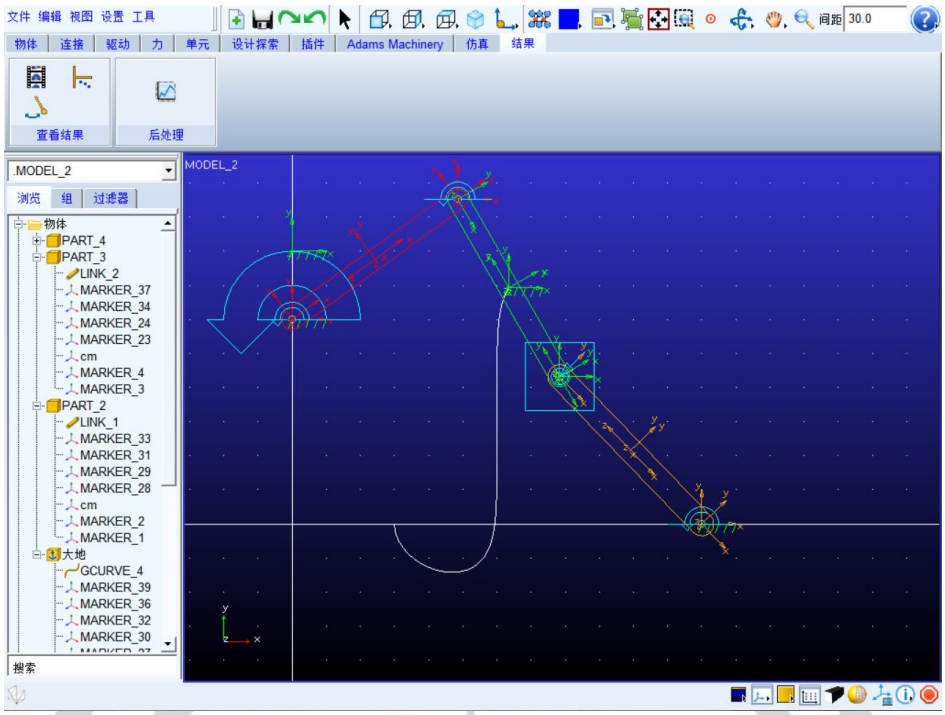


图 1.5.1.11　瓦特连杆机构在 Adams 软件中进行运动学仿真

运用 Adams 运动学仿真，构造如上图的试验机构，将杆长作为设计变量，将直线行程最大值作为目标值进行优化，可以发现，在一定的横向距离条件下，两边摇杆的长度与中间连杆的比值越大，则垂直上下的直线行程越大。最终，考虑到避震器行程与轮系空间大小，选定中间杆长$L2$和两边摇杆长$L1$、$L3$均为$60mm$的瓦特连杆机构，其中间连杆中心的部分仿真轨迹如下：

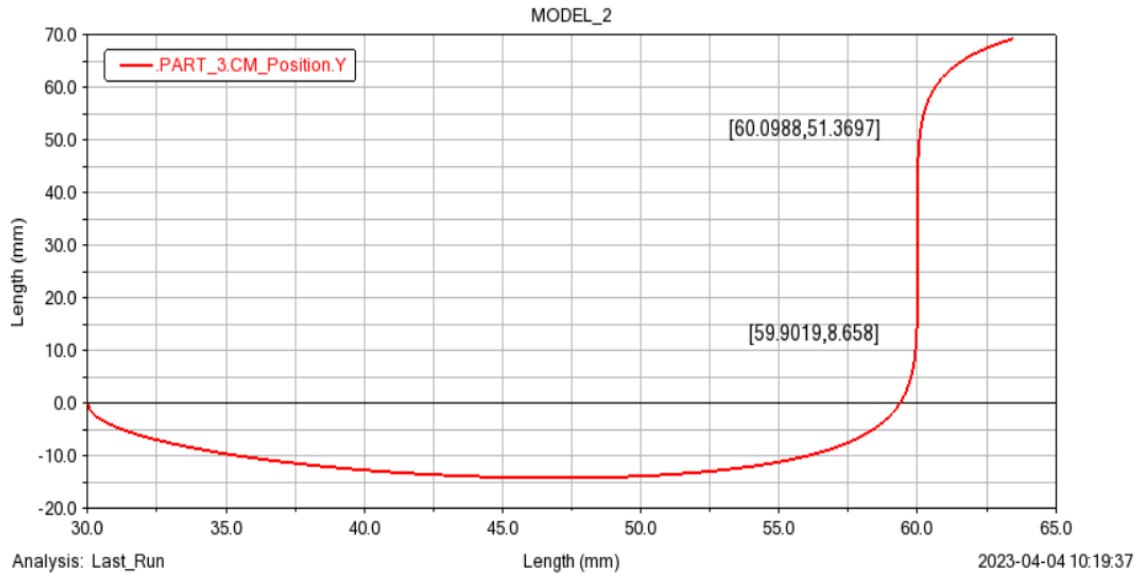


图 1.5.1.12　瓦特连杆机构部分仿真运动轨迹

由图可知，在中间连杆中心横坐标左右偏移不到$0.01mm$的情况下，竖直位移可以达到

$42.7117mm$，设计轮系时，将避震器未压缩状态设为接近直线行程最底端的状态，这样轮系压缩向上竖直移动可以达到约$40mm$的行程，既满足了我们轮系竖直方向位移行程的需要，又极好地保证了轮系的只有竖直位移，在横坐标上几乎没有分量，从而保证了轮距始终一致，提高了底盘的稳定性。

3.飞坡校验

在避震器的选择与设计上，主要考虑飞坡的需求。采用 Adams-Matlab 联合仿真，可以得到飞坡的各种位移、速度和力曲线，寻找在最佳飞坡姿态下的前轮所受冲击力，再进行悬挂压缩的仿真，可以得到悬挂的压缩情况，根据压缩情况选取合适的避震器参数。因此首先在 Adams 中构建简易的飞坡模型，并尽量保证关键参数与模型中一致（例如重心位置、轮径、车身关键尺寸、场地尺寸等）；又在 Matlab 中使用 Simulink 插件中设置输出位移和速度等曲线（D_x、D_y、D_z、V_x、V_y、V_z）如下图（图 1.5.1.13）

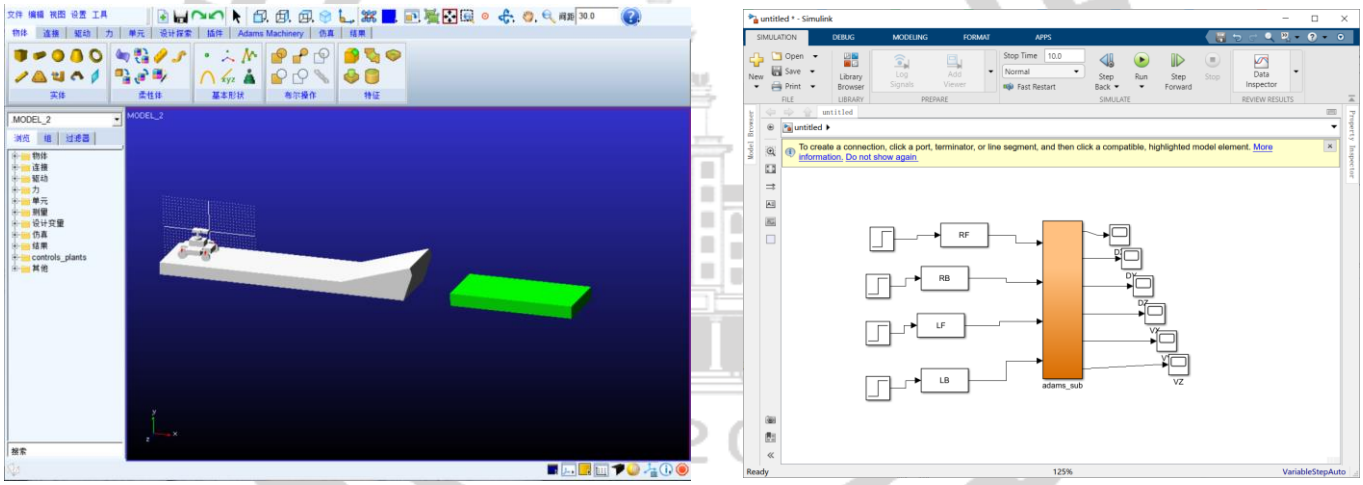


图 1.5.1.13　左：Adams 飞坡仿真动力学模型
右：Matlab Simulink 插件运行 adams_sub 模块输出曲线

在建立飞坡仿真模型时，主要是选取尽量接近真实的碰撞参数。在刚度系数的选择上，参考赫兹接触应力公式

$$\sigma_H = \sqrt{\frac{F_n}{\pi b}\frac{\frac{1}{\rho_1}+\frac{1}{\rho_2}}{\frac{1-\mu_1^2}{E_1}+\frac{1-\mu_2^2}{E_2}}}$$

再根据 Adams 的 Impact 函数模型，可得刚度系数

$$K = \frac{4}{3\pi\left(\frac{1-\mu_1^2}{\pi E_1} + \frac{1-\mu_2^2}{\pi E_2}\right)}\left[\frac{R_1 R_2}{R_1 + R_2}\right]^{\frac{1}{2}}$$

代入曲率半径、泊松比和弹性模量等值，计算K 约为$1418N/mm$,力指数一般选取1.5，阻尼约为K的0.1%到1%，穿透深度为$0.1mm$。在此参数下，不断调整哨兵机器人的水平运动速度,在$V = 3.4m/s$时，飞坡具有良好的落地姿态。

图 1.5.1.14　Adams 接触碰撞参数设置界面图

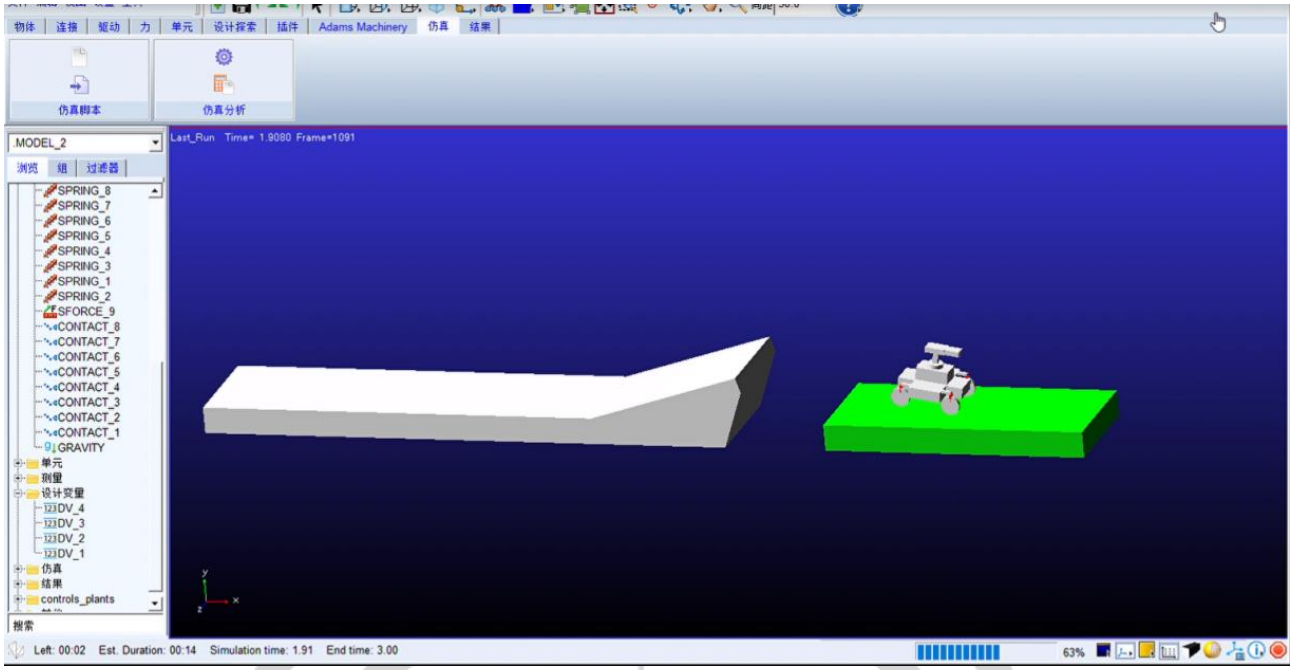


图 1.5.1.15　速度$V = 3.4m/s$时的飞坡落地姿态

在后处理中可以读取碰撞冲击力，再此基础上继续进行悬挂压缩情况的仿真。对轮系施加冲击力，设置避震器的刚度系数与阻尼系数，选取接近实际情况的值，观察悬挂压缩的情况。

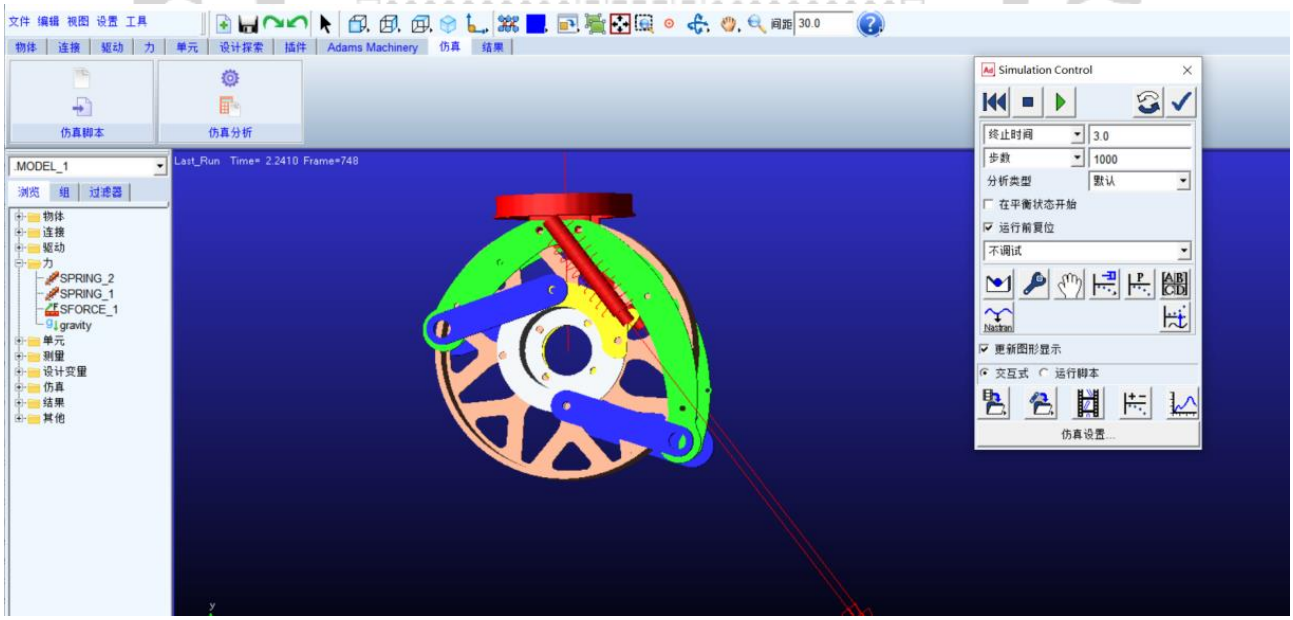


图 1.5.1.16　Adams 仿真悬挂系统受飞坡冲击力的压缩情况

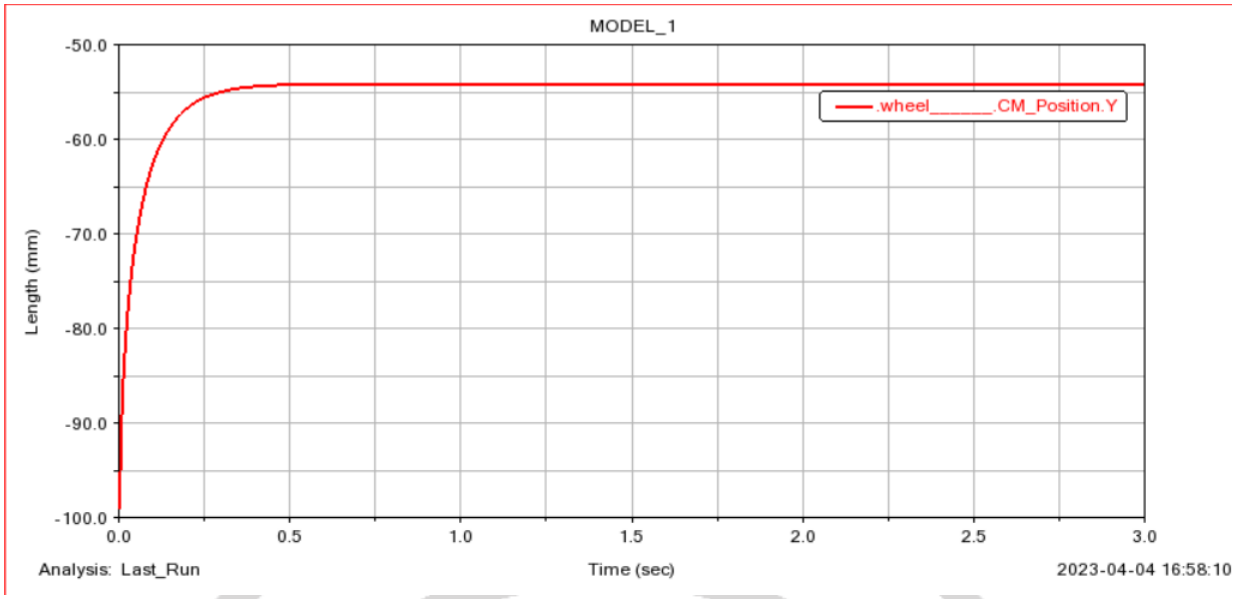


图 1.5.1.17 轮系悬挂压缩的竖直方向位移

由曲线图可以观察到在所设的悬挂参数下，竖直方向位移约为$44mm$，实际考虑到各连杆轴处的阻尼和防撞导轮的添加，悬挂的压缩量可能更小，因此上文所述的瓦特连杆悬挂设计校核通过。

4.急停刹车校验

本舵轮轮系舵向采用齿轮传动，在急停刹车时，前两轮向内转，后两轮向外转，形成八字刹车，这势必会对轮齿产生冲击载荷，还会对悬挂板产生一个大的弯矩。因此，需要对这两个零件进行单独的校核。于是，我们使用 Adams 进行刹车的动力学仿真，用 Solidworks simulation 对悬挂板进行静应力分析。同理首先，在 Adams 中建立刹车的动力学仿真模型（图 1.5.1.18）。

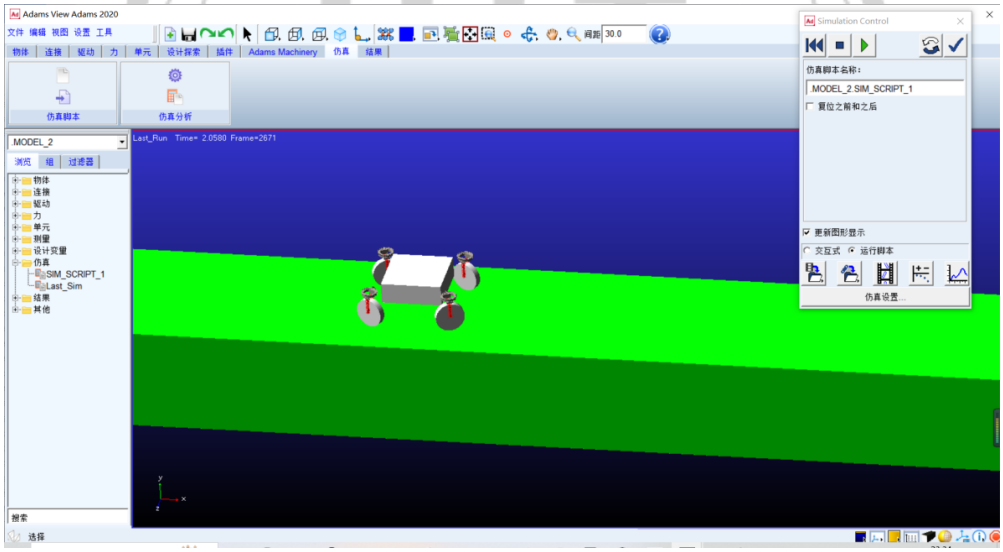


图 1.5.1.18 Adams 刹车仿真运行效果

在刹车的仿真过程中需要中途解除轮系航向的驱动，添加舵向的驱动，所以需要用到脚本仿真，而不能简单地直接仿真。刹车仿真的脚本命令逻辑为：在轮子转速达到所需的转速后，解除航向的转动副和驱动，激活舵向的转动副和驱动，舵向旋转一定的角度，从而使整车在摩擦力作用下刹车停止。另外由于齿轮传动在 Adams 中通过添加接触力实现，将两个齿轮均设置为钢的接触参数。在后处理中，我们可以测量得到四对齿轮传动的接触力曲线（图 1.5.1.19）。

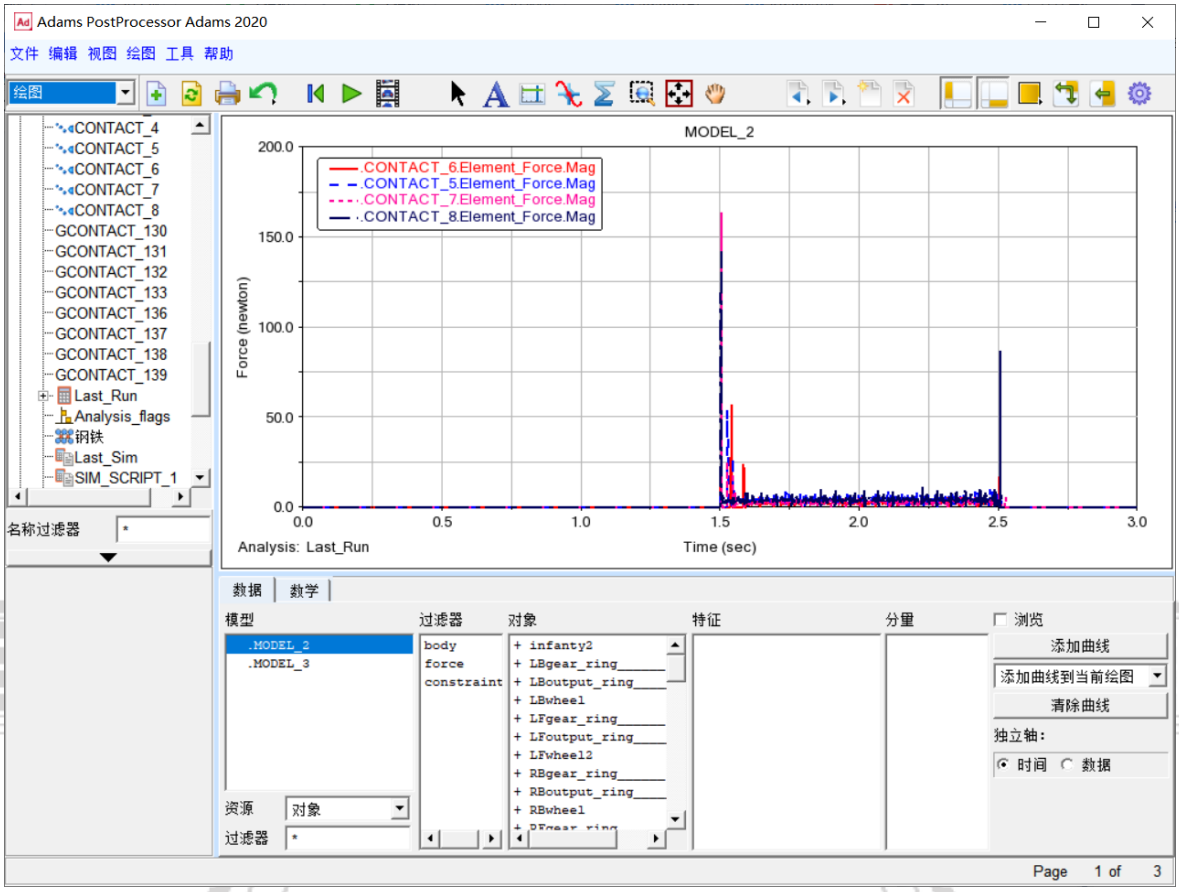


图 1.5.1.19　仿真刹车齿轮啮合力曲线

通过上图的曲线，测量得到最大的周向冲击力为$164.5N$。根据齿轮其他静力学参数简单计算可知，该处钢制小模数齿轮校核通过。

在后处理中，我们还可以得到轮子在 x 方向上所受最大冲击力为$940N$。对轮系的悬挂板可以简化为一个悬臂梁模型。其悬臂端的挠度与挠转角分别为

$$\nu = \frac{-\mathrm{Fa}^2(3\mathrm{L} - \mathrm{a})}{6\mathrm{EI}}$$

$$\theta = \frac{-Fa^2}{2EI}$$

悬挂板近似为矩形截面，其惯性矩为

$$I = \frac{bh^3}{12}$$

因此如果只需增大悬挂板的厚度，可以很大程度上减小悬挂板的形变。下面对悬挂板进行简单的静应力分析验证。利用 Solidworks simulation 插件，在悬挂板上施加 940N 的力，观察悬挂板的位移图。

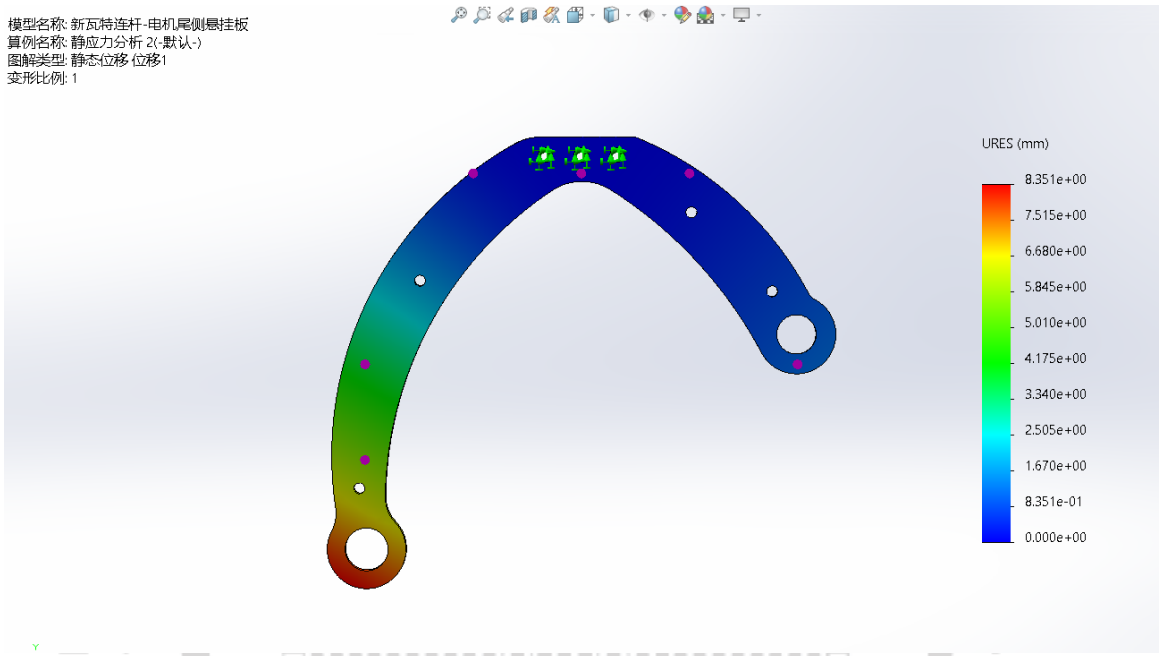


图 1.5.1.20　单悬挂板无加强板在刹车横向力作用下的静态位移

可观察到单个悬挂板的位移约为$8mm$，这是无法接受的。所以采用双端支撑，并为悬挂板额外添加小块加强板，使得局部截面厚度提高一倍，抗弯刚度为原来的 8 倍。重新在相同条件下进行仿真，可以观察到在添加加强板后位移明显减小（图 1.5.22）。

图 1.5.1.21　悬挂加强板布置形式

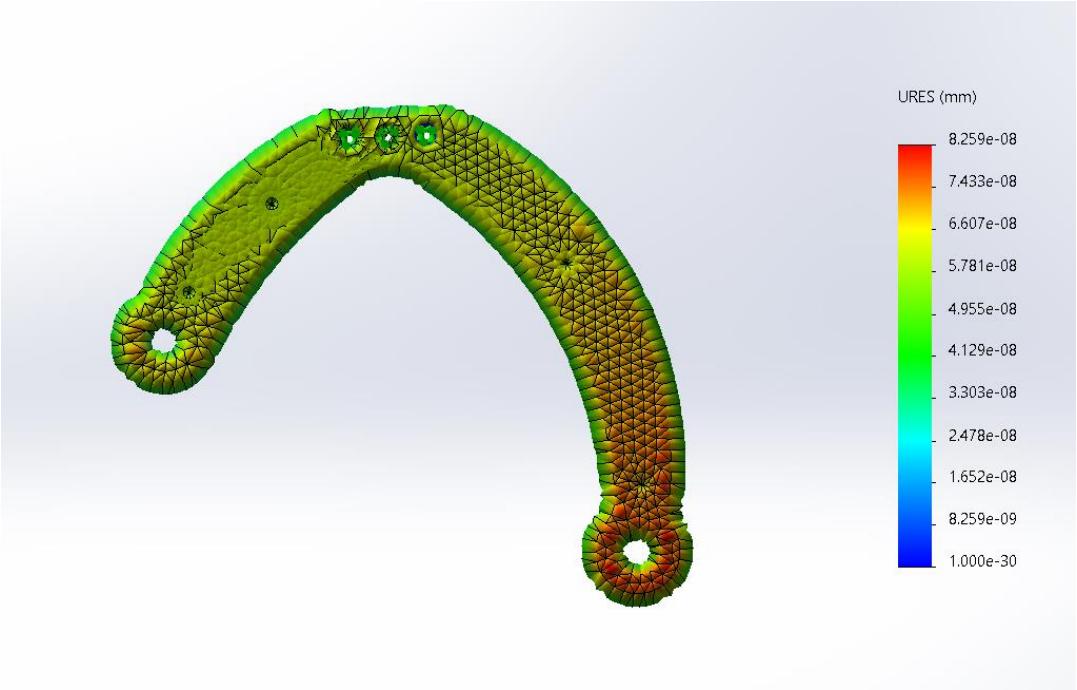


图 1.5.1.22　双悬挂板有加强板在刹车横向力作用下的静态位移

1.5.1.3 云台部分设计

一级云台部分

1.切枪机构

在本赛季中，虽然哨兵相对于往年有很大的改动，但是仍保留了双测速模块单独结算热量的特点。因此利用有限的空间实现快速切换测速模块非常重要。本设计设想通过平行移动的方式将刚性连接的两个测速模块进行位置的切换。

平移机构的导向采用的是平行双导轨（如图 1.5.1.23），本设计考虑到滑轨与滑块的链接并非刚性，即自由端受载后滑块和滑轨并非完全平行，而是有一个较小的挠转角，在悬臂的另一端则会产生一个较大的挠度（如图 1.5.1.24-（a）），导致枪管与测速模块的同心度大大降低。双滑轨结构则能很好的解决这个问题，即将两个滑轨滑块连接均抽象为铰接（如图 1.5.1.24-（b）），在铝管刚度足够大的情况下，自由端受载后产生的挠度可以忽略不计，可以很好地保证测速模块与枪管的同心度。

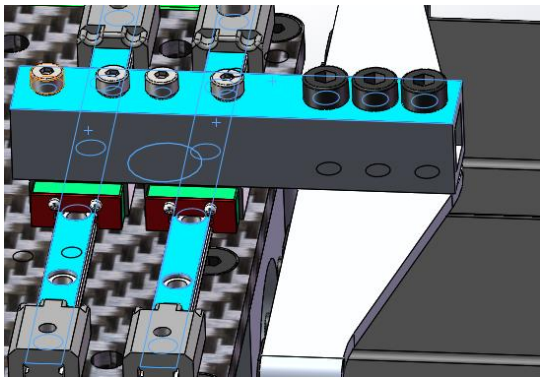


图 1.5.1.23　平行双导轨机构

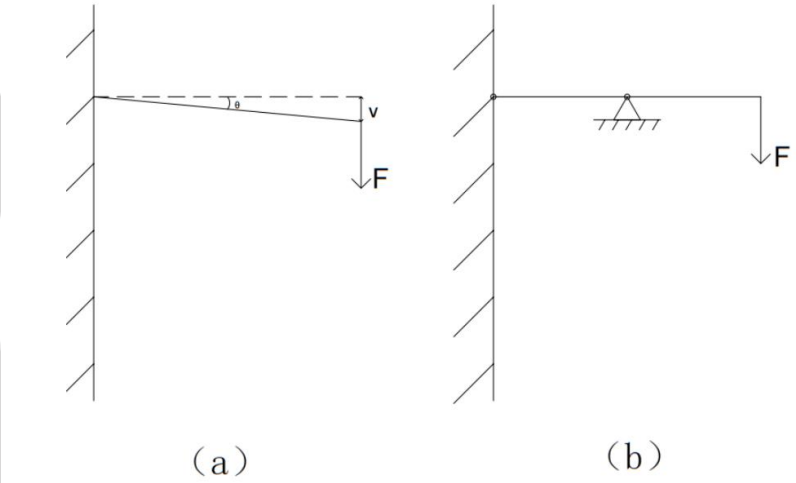


图 1.5.1.24　导轨结构示意图

为节省空间，我们的设计期望如下：

（1）只有一个动力源；

（2）尽可能减少平行于导轨方向的空间占用。

为实现上述期望，我们设计采用由舵机驱动的曲柄滑块机构，并将其安装在无用空间较多的云台下侧，如图 1.3。易知曲柄滑块机构的自由度为 1，因此其所需要的动力源数量为 1，符合期望。下面对相应的参数计算。

首先，官方测速模块的宽度为$43.4mm$，为方便计算，将两侧速模块的中心间距调整为$45mm$，即平移的行程为$45mm$。将测速模块抽象为滑块，测得舵机的舵盘中心与滑块垂直于导轨方向的距离为$44mm$，与枪管中心线垂直于枪管方向的距离为$60mm$，绘制草图如图 1.4。接下来要确定曲柄和摇杆分别的长度，令摇杆长度为L_1，曲柄长度为L_2，为防止定位点离死点过于接近，左右各给$7.5mm$余量。长度计算方法基于三角形两边之和大于第三边，两边之差小于第三边的原则。能列出下列两式:

$$\begin{cases} L_1 + L_2 \geq \sqrt{\left(60mm + 22.5mm + 7.5mm\right)^2 + \left(44mm\right)^2} \approx 100.2mm & (1) \\ L_1 - L_2 \leq \sqrt{\left(60mm - 22.5mm - 7.5mm\right)^2 + \left(44mm\right)^2} \approx 53.2mm & (2) \end{cases}$$

联立（1）（2），取$L_1 = 78mm$，$L_2 = 25mm$。

该设计不仅能可以利用无用空间，并且结构简单，只需要注意设计过程的准确以及关键点应远离死点的要点即可。

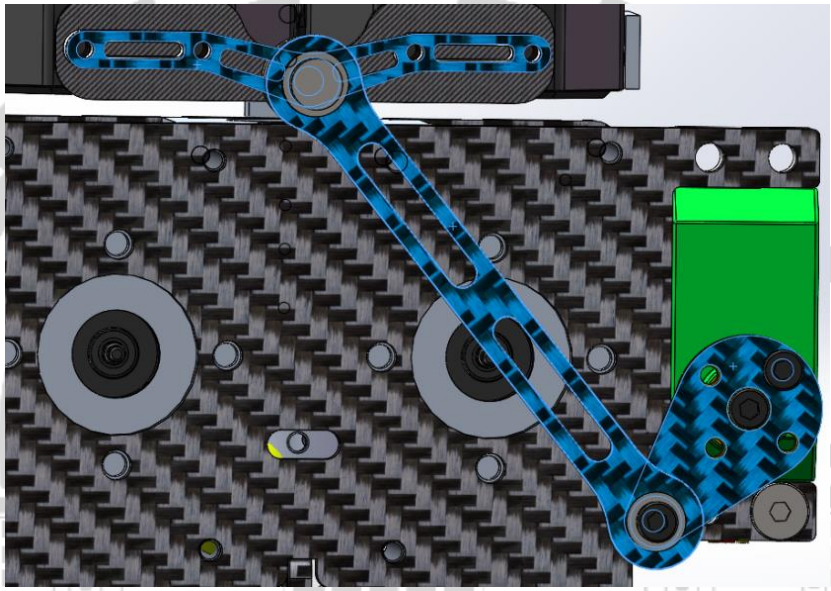


图 1.5.1.25　曲柄滑块切枪机构

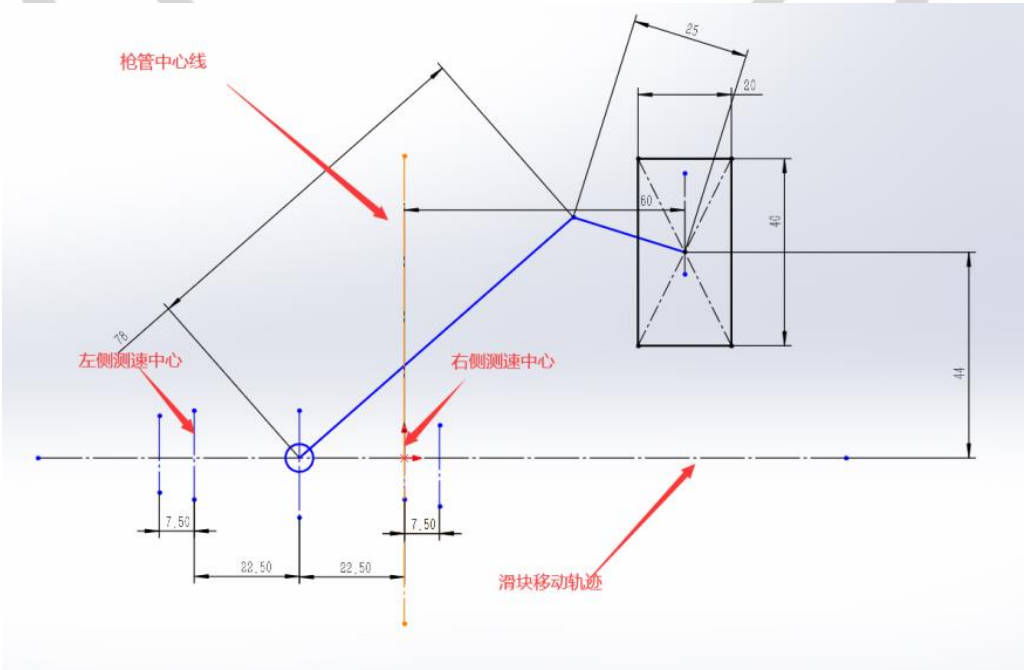


图 1.5.1.26　切枪机构草图

2.2006 发射机构

由于哨兵的特殊性，在云台的基础上还需要搭载一个二级云台来完成上层功能。因此，一级云台需要尽量地小巧，为满足这一需求，我们设计了相对于传统 3508 电机发射机构体积更小，重量更轻的 2006 电机发射机构，如图 1.5.1.27。

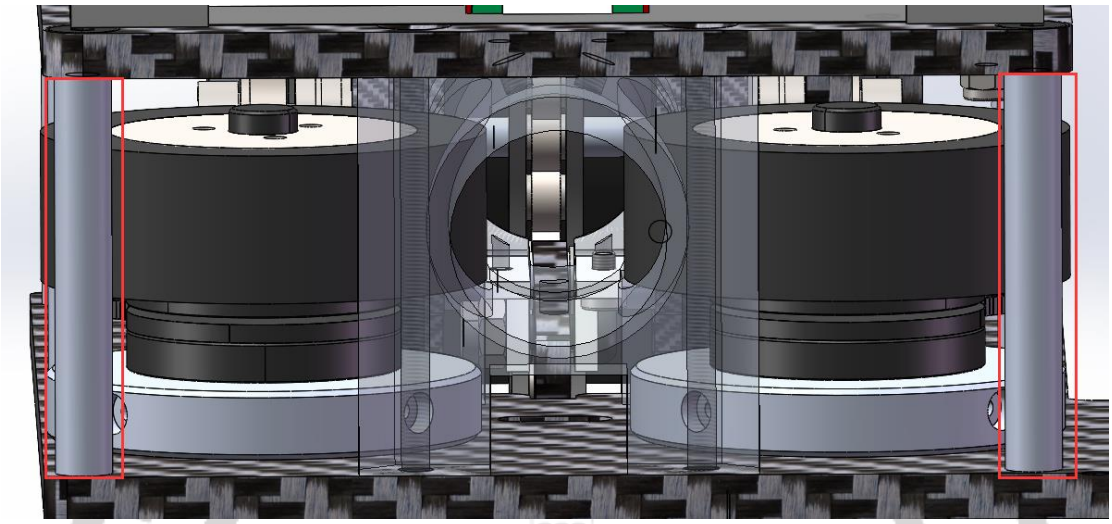


图 1.5.1.27　2006 电机发射机构

虽然 2006 电机发射机构相比 3508 电机发射机构能够节约很多体积和重量，也正由于 2006 电机转子直径更小，想要让$17mm$弹丸达到$28\sim29m/s$的速度所需的电机转速更高。但安装电机的碳板刚度不足时，碳板振动的的固有频率和电机转速相同，会产生共振，大大降低机构的稳定性。因此，为解决可能产生共振的问题，我们放弃了传统步兵发射机构的单板悬挂式的电机安装方式，而是改用如上图的双板刚性连接的方式（图 1.5.1.27 红框处），提高了固定电机碳板的刚度，从而大大提高了其固有频率，有效地避免了结构的共振。

3.下供弹链路

由于哨兵拥有 750 发$17mm$弹丸的发弹量，像步兵一样的上供弹模式会导致 yaw 轴偏心过于严重，且 pitch 轴控制所需的补偿也更多。因此，选用类似英雄机器人的下供弹模式。而对下供弹来说，由于供弹链路需要从底盘一直延伸到枪管进弹口端，这无疑会大大增加卡弹的风险，关键在于解决进弹口处的过渡处理。处理方式如下：

首先，向上供弹到枪管进弹口端的过程有一个接近90°的转弯角度，通俗的说弹丸从竖直向上移动到水平射出的过程需要在链路中完成。如果在采用直角转弯的处理方式（图 1.5.1.28），可以看出，弹丸②给弹丸③提供的推力几乎竖直向上，其水平分量很小，同时弹丸③给弹丸④提供的推力也有一部分向上的分量导致弹丸④与链路壁产生压力从而产生阻力。因此直角转弯的方式极易在转弯处发生卡弹。

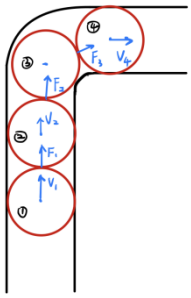


图 1.5.1.28　直角链路中弹丸受力图

为了防止卡弹，我们期望链路应该曲率越大越好，但是曲率过大又会导致链路链路的出弹口无法水平出弹。这时，考虑到我队哨兵设计的技战术要求是可以用来击打环形高地并且在五米内可以迅速击杀敌方机动地面单位，我们选择仰角10°的角度代替水平位置作为平衡位置，即链路的出射角度设置为10°，如图 1.5.1.29（b）。同时为了满足 pitch 轴的转轴和 yaw 轴中心投影重合，即链路的进弹口和出弹口在的水平距离尽可能接近，选择两段曲率相等的圆弧作为链路的基准线，如下图所示链路形状类“S”形。在链路和枪管的连接处则参考我队去年英雄机器人的滑槽导板，保证在最大俯仰角区间内弹丸不会被卡在连接处。在测试中，我们设计的链路可以在允许的俯仰角区间内持续发射弹丸 10 分钟并未发生卡弹，效果很好。

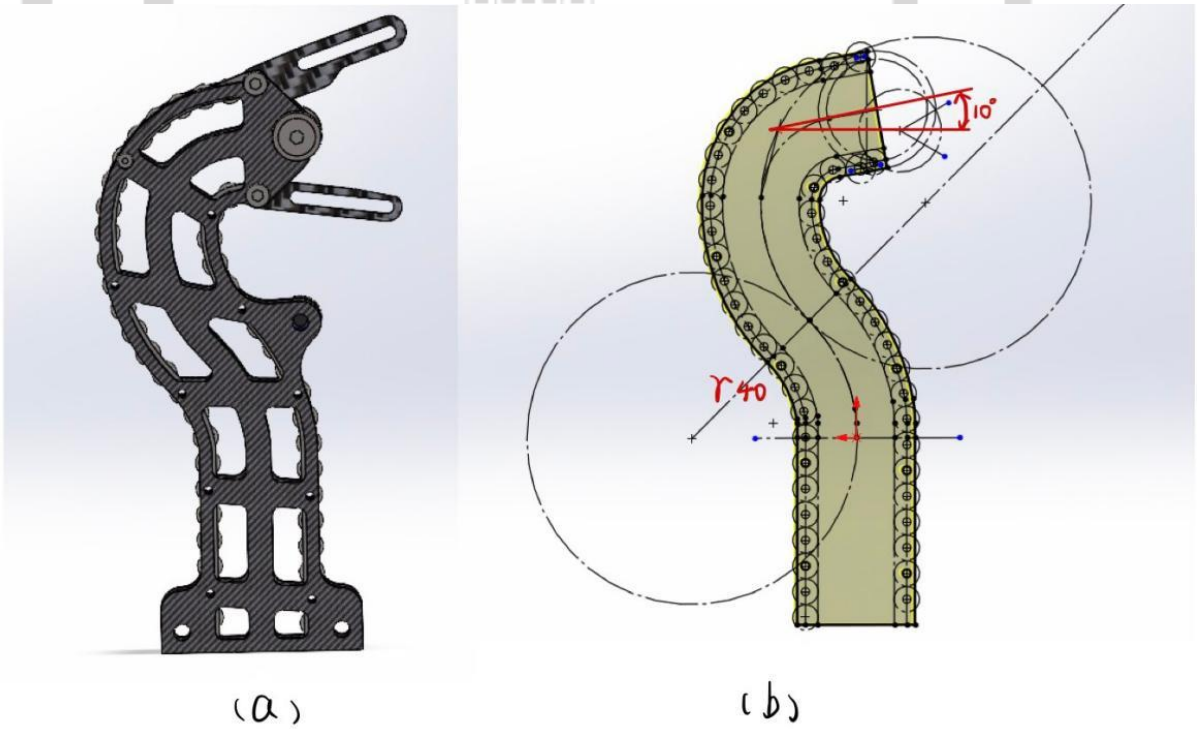


图 1.5.1.29　哨兵机器人链路示意图

自研滑环

滑环是会发生相互旋转的机构间电气连接的重要部件。由于哨兵下供弹的特性以及当前滑环的标准型号，只能选择电机云台非直连，走滑环中轴过弹；或者电机云台直连，滑环外

套电机中轴过弹。前者需要多加一个传动装置，占用空间；后者节省空间，但是市面上买不到相应尺寸的滑环。为了提高机器人的上限，参考南京航空航天大学祝老板的未出版滑环模型，我们选择对滑环进行自研。

首先，我们对于滑环有以下要求：

（1）滑环中孔应大于 6020 电机外径$66.7mm$

（2）滑环的高度应尽可能接近 6020 电机的高度$42mm$

（3）线数为 9 粗 9 细，粗线最大载流 10A，细线最大载流 3A。

（4）连接稳定，电气属性良好。

滑环工作的原理就是外圈通过刷丝与导体圆环导通，同时导体圆环与内圈连接，从而保证内圈在转动的过程中与外圈始终有良好的电气连接（如图 1.5.1.30）。

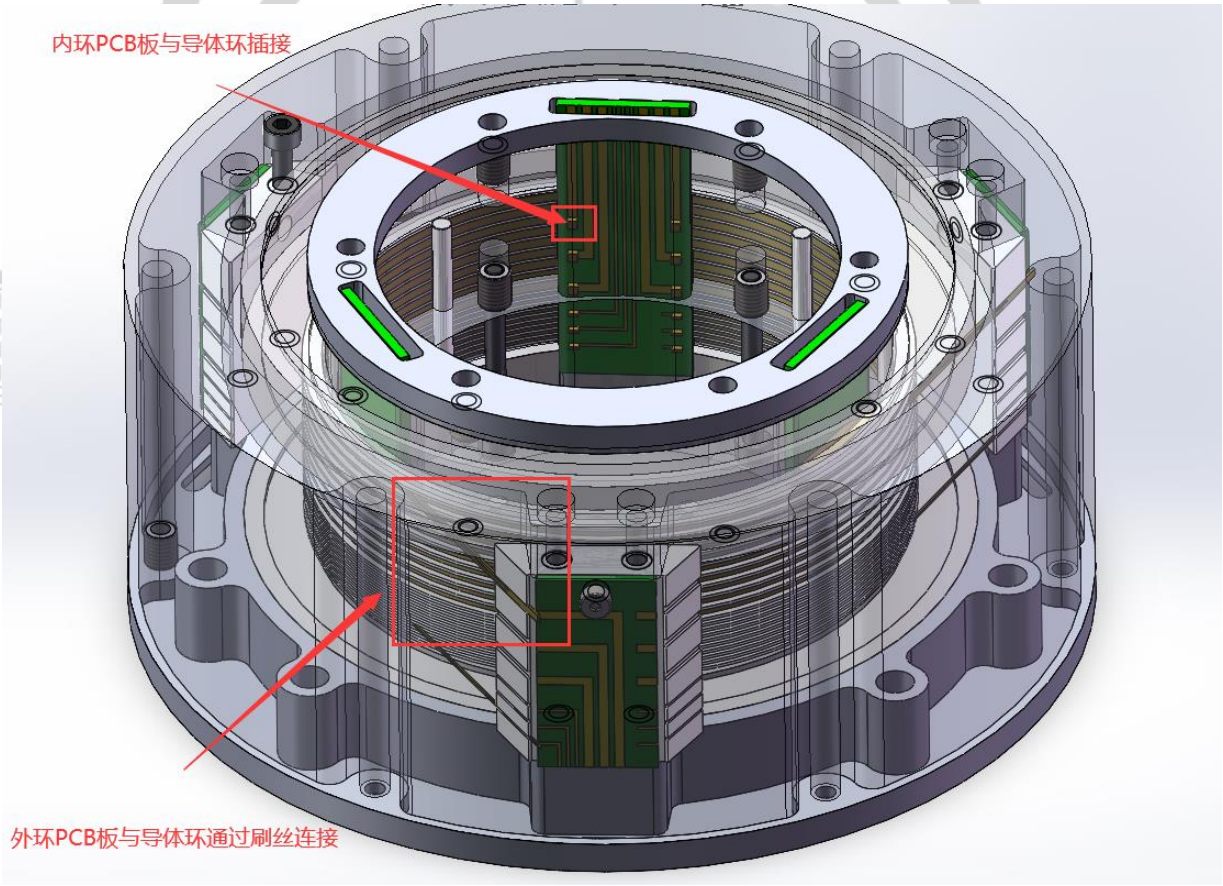


图 1.5.1.30　初代自研滑环 模型图

滑环和刷丝是导电滑环最重要的两个组成部分。其主要性能是：要求材料导电率高，导热性能好，熔点高，电阻系数小；具有抗氧化性，抗腐蚀性等特性；具有一定的硬度，耐磨性好等特点[25]。结合以上特点，我们选用黄铜来作为滑环材料，并且在其表面进行镀镍处理，

使其具有抗氧化性和抗腐蚀性。设计过大电流的圆环的最小截面为$1mm \times 1.5mm$，过小电流的圆环最小截面为$1mm \times 0.5mm$。由于其工作表面为侧面，因此侧面采用粗车-精车的加工工艺，而由于环厚度极薄，切断时加工工艺不能选用切断车刀，选用线切割的方式在精车后分层切薄片。

刷丝可以选用金镍丝，但是金镍丝价格昂贵，可以选用黄铜在其表面镀一层钯，可以有良好的防腐作用[26]。因此，我们选择黄铜作为刷丝，长度尺寸为$50mm$，最小截面分别为$0.3mm \times 1.4mm$和$0.3mm \times 0.4mm$。

在实际制作过程中，测试发现黄铜这个材料在所要求的尺寸下刚度低韧性差，无法提供所需的预紧力压在铜环上，从而导致连接非常不稳定。并且刷丝对滑环的压力越小，刷丝和滑环的接触电阻就越大[25]。导致过流时接触点电阻过大，使刷丝发热升温，这样会大大降低滑环的寿命。因此，无论是从滑环的稳定性还是寿命而言，黄铜这个材料都是失败的。

第一次失败之后，我们把更多的注意力放在寻找刚度大或韧性好的材料上。从韧性的角度想到了记忆合金（镍钛合金），这个材料具有优异的超弹性、形状记忆效应，对其进行一次完整的加载-卸载循环，卸载后合金能够完全恢复到变形之前的形状[27]也就是说，当给记忆合金加载使其弯曲变形之后将其压在滑环上之后卸载，它就会有一个持续的回复力直到回到加载前的状态，这个回复力就可以提供所需要的预紧力。在实测过程中，其预紧力确实能达到要求，并且连接的稳定性非常高。但是当其过流时，仍然会持续升温，温度甚至能达到$100℃$以上，并且在这个温度下记忆合金发生了形变，因此该材料也未通过测试。最终我们发现原因在于其本身的电阻相对于通路中的其他部分而言高了一个数量级，由于单位时间发热量$Q = I^2R >$单位时间散热量，热量会持续累积，产生了上述问题。

经历了第二次失败，我们改变了选材策略，从导电性能良好为第一要点出发，结合合金的刚度大于纯金属的理论，去寻找材料。为了节省经费，我们仍然以铜为基材，从市面上最容易买到的白铜（铜镍合金）开始测试。最开始我们选用的时白铜丝作为刷丝，粗线直径为$1.4mm$，细线直径为$0.3mm$。粗线测试过流 6A，非常轻松地就通过了测试，连接稳定并且刷丝并没有升温，由于受到示波器过流的限制，最高只能测试到 6A，预测过流 10A 也可以通过。但是细刷丝仍然遇到了上一次一样的问题，持续升温。但是我们测量后发现，其阻值并不算高，因此我们认为应该是细刷丝侧面积太小，散热性能太差。之后，我们将细刷丝改为了白铜片来增大其散热面积，最终也成功通过了测试，在 3A 的过流下没有升温。就此我们完成了完成的初代自研滑环设计（图 1.5.1.31）。

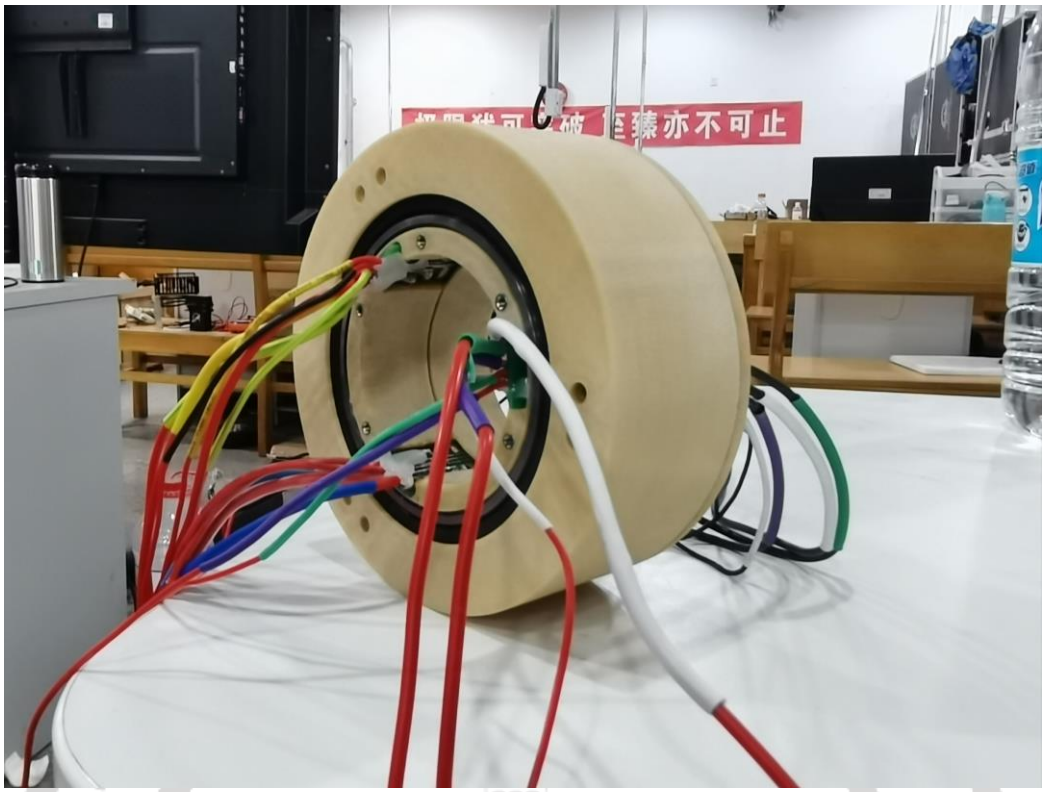


图 1.5.1.31　初代自研滑环 实物图

目前该滑环的改进方向为：改 PCB 板为接插件，增加硬件接线稳定性与便于走线。滑环内部的刷丝形状进一步修改，同时兼顾加工方式与刷丝的刚度保障。

二级云台部分

二级云台在本文中指由于上层需要安置激光雷达及全局视野相机等传感器，并且不希望该部分传感器与一级云台发射机构固连，因而独立出一个相对世界坐标系不存在旋转变换的新云台。设计该新云台的原因为发射机构需要有较快的响应，因此在正常工作时经常处于高加速度的状态。这样的加速度会对激光雷达传感器性能及相机的抓图效果产生影响。

1.需求分析

首先二级云台独立于其他主机构部分，该部分只含有一个自由度且处于全车最高处，因而必须保证该系统自身刚度与该系统连接的接头刚度足够。其次，该部分需要包含 2 个运算平台、1 个激光雷达、3 个相机，因而在这些固定质量之上越少增加结构性质量（为了固定传感器等固定元件所需要增加的结构的质量）越好。

2.结构设计

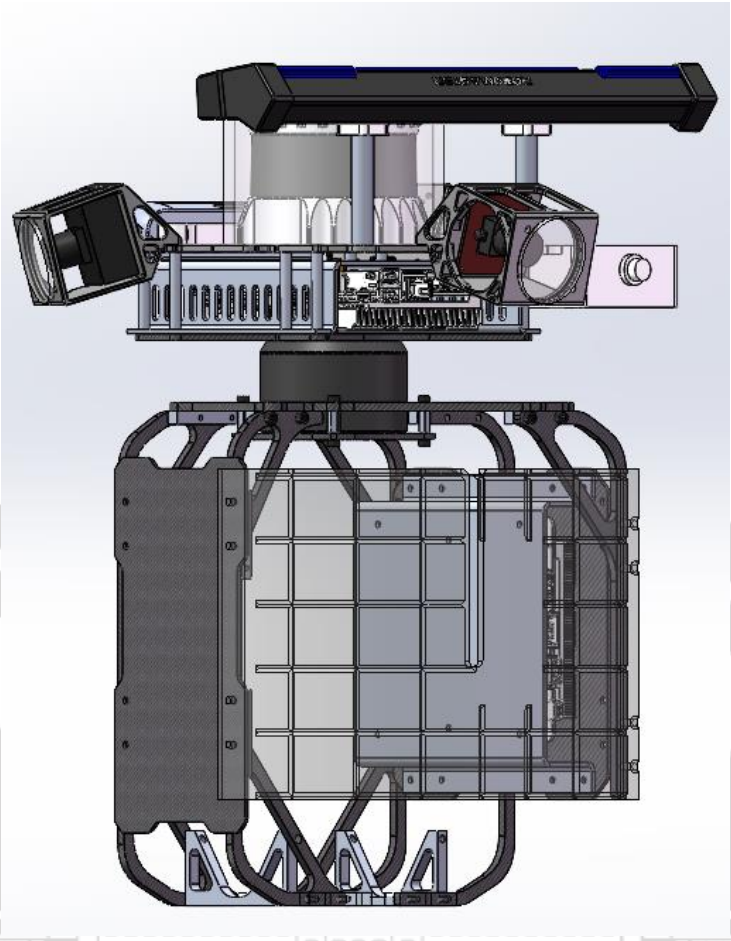


图 1.5.1.32　二级云台总成 模型图

二级云台模型见上图。上半层与电机转子相连，利用双层板夹一台定制钣金保护壳的 NUC11 实现第一个运算单元的布置；顶板单块板子完成激光雷达、三个全局视野相机、灯条、控制盒等元件的安装和固定。钣金件外侧固定孔设计为与直角连接件等尺寸的标准尺寸孔，方便拓展。下半层为了与一级云台分开的同时轻量化，主结构使用了四条厚为$4mm$的 C 型骨架支撑，并上下表面都通过直角连接件连接到底盘与电机固定板。C 型骨架可以保证结构在垂直方向上有着极高的刚度，碳纤维本身的材料特性也保证该机构不会出现失稳的问题。但存在的结构问题便是在水平方向的横截面积过小（约等于$4*4*6=84mm^3$），因此解决方案是使用整块薄碳板连接单侧骨架，并用曲面打印件将后面连接。使它在横截面上也变成 C 型。与此同时侧板也可充当保护的作用，主梁也方便电控布线。

在实际装配时，发现主梁及以上部分的连接刚度已经足够，但下方与 yaw 回转加工件的连接并不稳定。承受横向载荷时，整个机构是一个悬臂梁机构的变形。只能通过增加连接点的形式来增强下方的连接刚度。但这样的方法并不能从根本上解决问题。下一代的设计设想是，将图 1.5.1.29 - (a)中的链路侧板直接向上做出结构连接到二级云台回转电机的固定板中。

这样的好处是将二级云台的受载分散给一级云台一部分，从而提高整体的刚度。

1.5.2 硬件设计

1.5.2.1 硬件设计综述

哨兵机器人的硬件系统包括主控制板，云控制板，超级电容模块，超级电容组，超声波扩展板，荧光充能模块等。

在控制方面，主控制板，云控制板是机器人底盘控制和云台控制的硬件单元，同时，云控制板还能够提供九轴陀螺仪数据。

在功率方面，超级电容及超级电容组实现了在功率限制的情况下，对能量零存整取的效果，在实际使用时，将超级电容串联在电源和底盘设备之间，超级电容模块的性好坏能是整车机动性的关键。

另外，设计了超声波扩展板，接收多路超声波信息以实现底层防撞保护机制，设计了荧光充能模块，采用铝基板在保证 LED 输出功率的同时尽可能提高散热能力。

总地来说，在硬件设计时，采用的核心原则为在实现功能及保证稳定性的基础上，尽可能地方便电控布线和降低模块成本。硬件系统主要单板选型及评估情况如下表所示。

表 1.5.2.1 单板硬件选型与评估

单板	设计需求	风险评估
主控	提供充足的外设接口，具有一定保护功能	偶尔出现无法复现的 CAN 通讯失能
云控	与购买相比降低单板成本，具有一定保护功能	陀螺仪标定校准及温度变化导致解算 yaw 角度漂移
超级电容	储存能量，有快速充放能力，高效率	大电流模块寿命有限，出现损坏对整车机动性是致命打击
超声波扩展板	接收多路超声测距信息实现底层防撞保护机制	基本无风险
荧光充能	与购买相比降低单板成本	基本无风险

整车的硬件系统整体框图如下图所示：

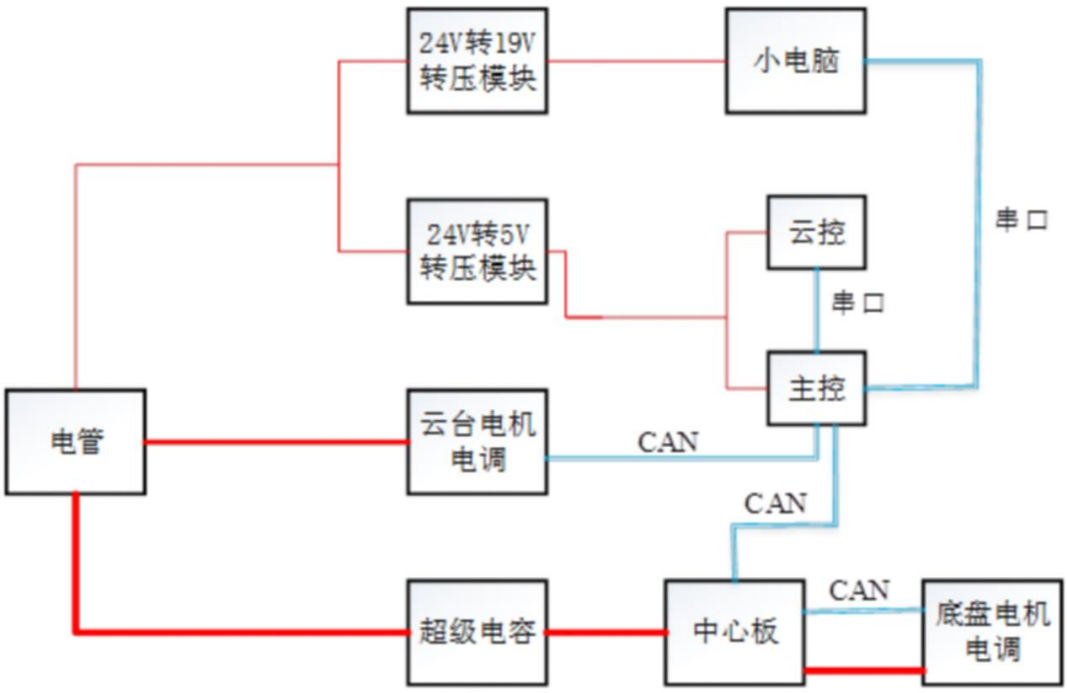


图 1.5.2.1　硬件系统整体框图

下面针对硬件设计部分一些研发项目进行说明。

1.5.2.2 超声波测距板设计

设计背景

哨兵机器人是场上唯一的的自主机器人，为了到达指定地点执行任务，建图，避障，路径规划等工作必不可少，由于哨兵机器人使用的激光雷达垂直视场角为 90°，水平方向向下扫描的角度为45°，且水平方向激光光束较为密集，靠近两侧激光光束较为稀疏，同时车体上激光雷达布置较高，这也就导致激光雷达对车身附近的较矮小障碍如台阶或掉落的矿石的实时感知存在一定盲区，因此，我们通过车身上布置的多个超声波传感器进行融合感知，便于进行实时碰撞检测及路径优化。

原理图、PCB 及外壳设计

该超声波测距板选用 STM32H743IIT6 芯片，引出了 8 路超声波适配的端口，8 路串口、2 路 CAN，3 路 PWM，3 路 SPI，2 路 IIC，并在板子上集成了 USB-TTL 电平转换芯片，此外预留了网口、USB、SD 卡等备用接口。该超声波测距板的原理图和 PCB 如下图所示:

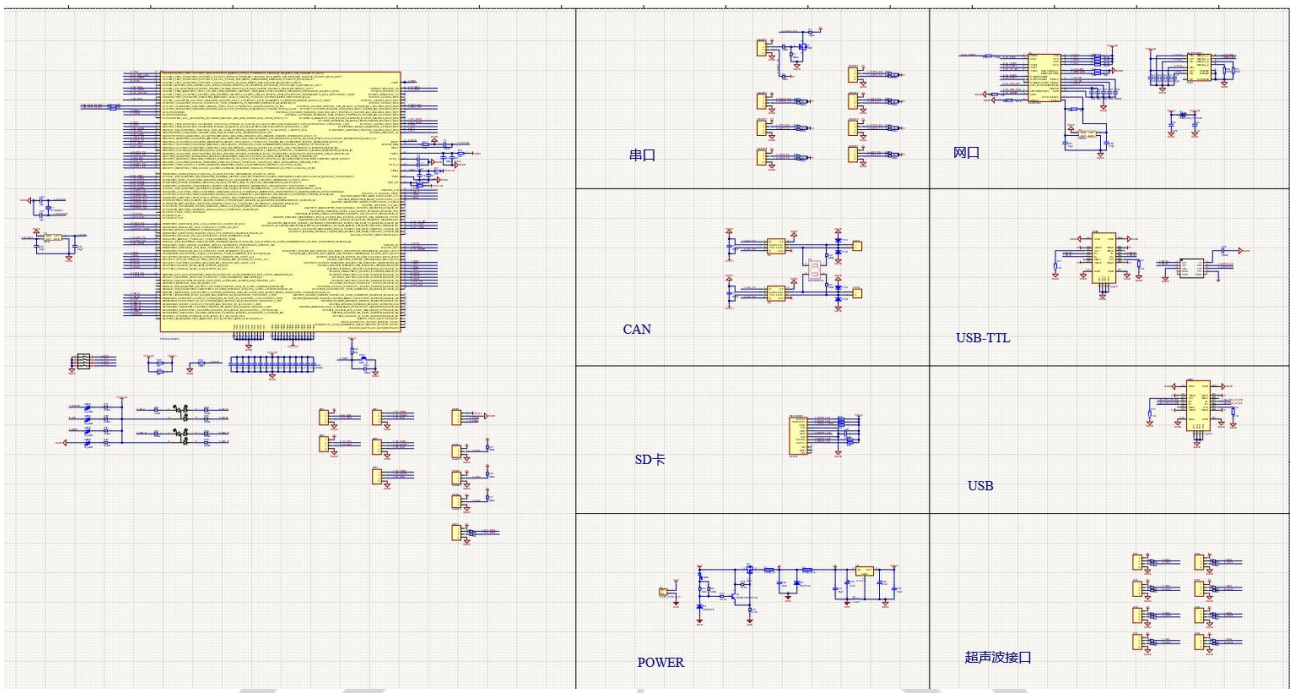


图 1.5.2.6　超声波测距板原理图

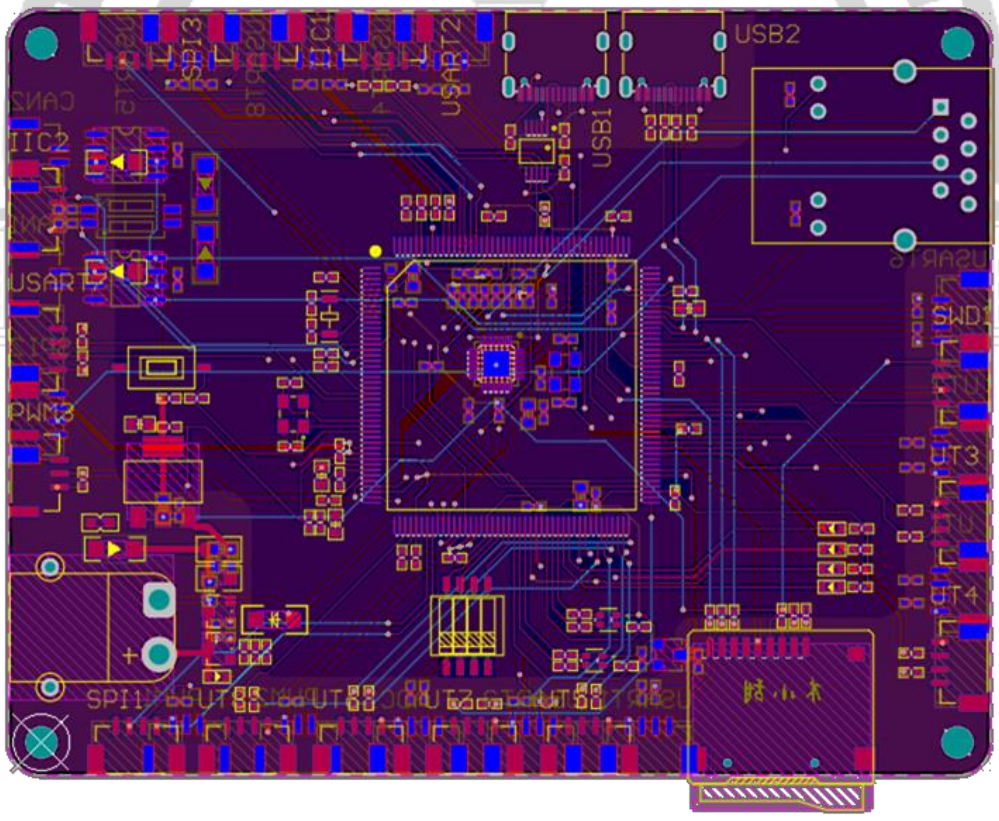


图 1.5.2.7　超声波测距板 PCB

设计说明

超声波测距是一种常见的测距方式，其原理是利用超声波在空气中的传播速度，通过发射超声波并接收其回波，计算物体与传感器之间的距离。

与此同时，STM32 具有高精度的定时器捕获功能，可以实现对超声波信号的精确测量和计算。在具体实现上，我们首先配置定时器为输入捕获模式，选择在定时器的输入捕获通道上连接超声波接收端口，当接收到超声波回波时，定时器将自动记录当前计数器的值，并停止计数。然后通过计算记录的计数器值和定时器的时钟周期，就可以得到超声波传播时间。根据超声波在空气中的传播速度，就可以得到物体与传感器之间的距离。

值得注意的是，为了保证测距的精度和稳定性，我们同时对超声波信号进行一些滤波和校准处理。采用了数字低通滤波器对超声波信号进行平滑处理，去除掉一些高频噪声和干扰。以提高测距的准确性和可靠性。

1.5.3 软件设计

系统架构

系统整体架构：

哨兵机器人软件部分的整体架构如图所示，顶层软件主要部署在两个 Mini PC 端，分别称为一级云台电脑和二级云台电脑，软件整体分为 3 层：决策层、功能层和驱动层。其中决策层用于汇总信息决定机器人行为，功能层负责实现感知、定位和规划的子功能(蓝色部分所示)，驱动层(绿色所示)用于从传感器和其他嵌入式设备中(橙色部分所示)获得传感器信息。

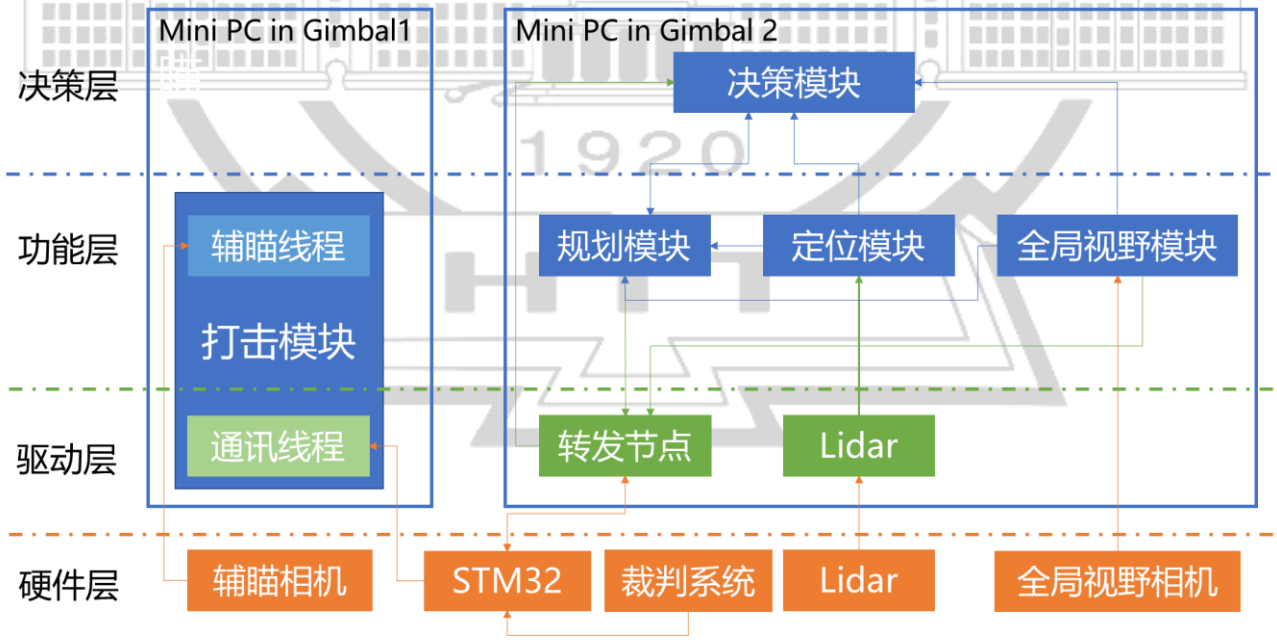


图 1.5.3.1　上位机软件架构

将软件部署在两个云台的原因是出于对处理器瓶颈和整体方案的考虑，主要有以下几个考虑要点：1. 辅瞄需要实现较高的帧率(200fps)，程序需要占用大概 60-70%左右的 CPU，出

于前期对全局视野模块和定位模块数据量和算力要求的保守估计，需要分摊到两个电脑以保证系统稳定性。2. 由于难以衡量机器人运行中的冲撞、快速运动对定位的影响，机器人设计了两级云台以保证激光雷达相对初始位姿基本不变。为了减少经过滑环的数据量和线数，因此在二级云台上放置了 Mini PC。

一级云台电脑如图左边框图所示，主要用于云台的控制，只运行一个辅瞄进程，设计运行帧率在 200Fps 左右。打击模块中包括打击目标决策、图像处理、串口通讯、目标运动估计和指令生成功能，主要有辅瞄线程和通讯线程，使用串口通过 STM32 转发实现和二级云台电脑的通讯。二级云台电脑实现主要的功能，各个模块的设计运行帧率在 10-30fps 不等。由于不需要高速传递大量数据，出于解耦考虑，二级云台各个模块直接使用 ROS 进行通讯。

全局视野软件设计：

全局视野(感知)模块基于上赛季哨兵机器人代码解耦开发，负责实现机器人周围视野 0-5m 内的全向感知功能，完成对于敌方机器人和我方机器人的位置检测功能，用于控制云台大致朝向辅助一级云台相机找到打击目标和决策。由于全局视野模块需要 3-4 个相机同时进行检测并将各个相机的检测结果融合，其量测更新速度无法用于实时控制，因此设计运行帧率在 10-30fps 以降低资源和算力的要求。(重点功能)

另外，出于具体进度的考虑，当前的全局视野模块中涵盖了部分打击决策，用于获得周围所有目标中最优的打击目标，并会通过转发节点发送给一级云台电脑实现具体打击的控制。

全局视野运行流程如下：1. 从参数文件中读取参数并初始化静态参数变量 2. 根据参数变量实现相机类和任务类的实例化，注册相机回调函数，初始化交互和通讯线程 3. 开启相机抓流、运行交互和通讯线程。在相机回调中实现对应任务类的检测、关联，并将检测得到的目标结果写入一个带锁的全局 Vector 中；通讯线程每隔 30ms 读取一次全局 Vector，获取最新的目标检测结果并进行发布；交互线程负责通过命令行实现快速的调试和可视化功能。

由于全局视野需要多个相机，为了方便适配不同相机的特性可能需要在不同相机上运行不同的检测算法，并将检测结果写入一个公共变量。出于上述考虑，全局视野采用“抽象类+继承类”的方案，采用了装饰设计模式，将每个在相机回调线程中运行的函数抽象为任务类 Task，针对神经网络检测和传统形态学检测分别创建了衍生类 TaskNN 和 TaskCV，在回调中使用智能指针调用实现运行时多态。同时在 Task 类中加入带锁的静态向量作为所有任务实例的共享向量，通过写入该 Vector 实现结果的汇集。

辅瞄软件设计：

基于队伍去年对各个兵种代码的整合，辅瞄程序基本软件结构和去年差别不大，主要通过维护和继承一套公共的视觉开发库 RM_CV_Base 实现功能的解耦和快速开发。RM_CV_Base 的整体架构如下，该库设计初衷为提供常用接口的抽象和一套默认的实现，不同兵种可以通过继承并重写少量的函数即可实现不同需求的验证和开发，经过验证优化后的代码也可以推送到库中实现快速的部署，实现所有兵种性能的优化。

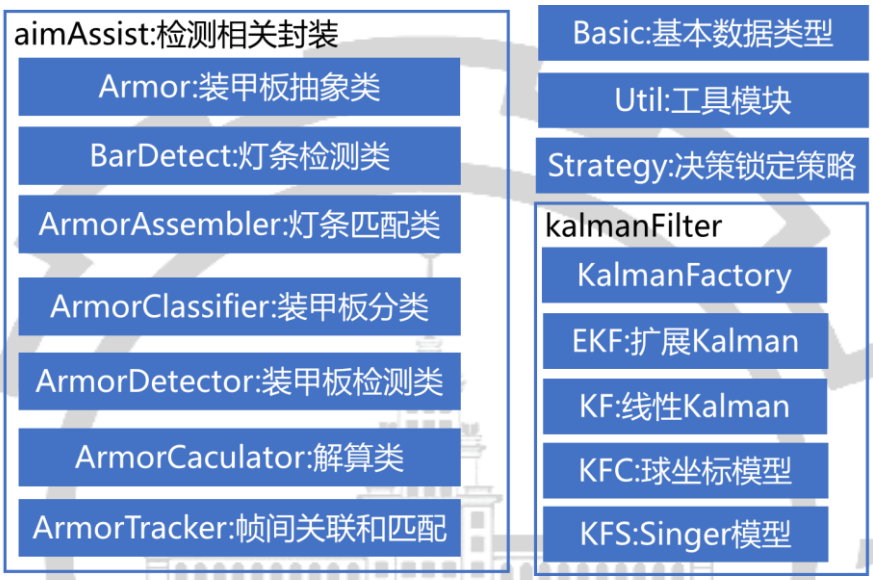


图 1.5.3.2　RM_CV_Base 架构图

运行流程

上述公共库基本涵盖了 RoboMaster 辅瞄的大部分功能，整体运行流程如下图所示，1. 首先通过相机回调和串口接收线程获取相机数据和 IMU 数据，进行时间戳软件对齐后写入 cv_base 提供的图像帧数据类型的全局静态变量。2. 将图像帧送入识别器和跟踪器，对应 cv_base 的 aimAssit 模块，识别器调用 ArmorDetector 的检测接口，具体检测流程为：先通过 BarDetector 检测灯条，然后使用 ArmorAssembler 实现潜在装甲板的灯条匹配，并使用 ArmorClassifier 实现二次校验剔除误匹配，最后使用 ArmorCaculator 实现装甲板位姿在相机坐标系下的解算和车体坐标系变换；检测器使用 ArmorTracker 模块进行具体的帧间关联，获取装甲板历史 N 帧的轨迹，用于进行估计。3. 预测器则通过对每个装甲板轨迹的丢失计数、最新量测进行目标是否存在的判断以及状态估计，状态估计部分调用 KalmanFilter 模块，利用工厂类创建滤波器对目标状态进行估计以获取目标的运动学参数，使用运动学参数预测经过子弹飞行时间后目标的位置，然后传入控制器进行弹道解算和补偿，将预期 Pitch 和 Yaw 发送给下位机。

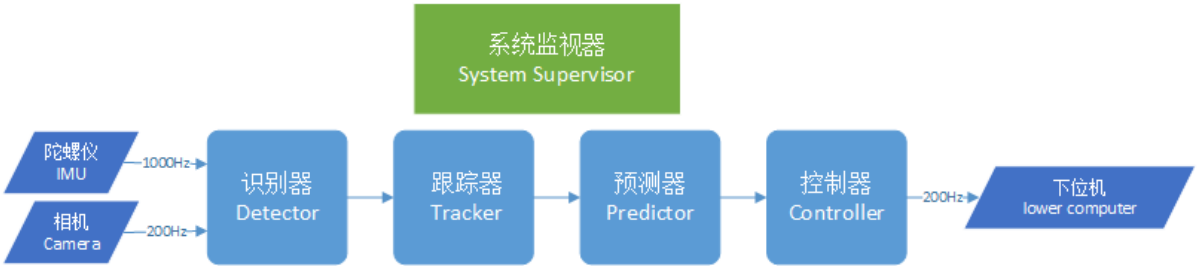


图 1.5.3.3　辅瞄运行流程图

ROS-FreeRTOS 的混合软件系统设计

哨兵机器人的软件系统由 ROS 机器人操作系统和 FreeRTOS 实时操作系统共同搭建而成。根据所执行任务的复杂度高低和及时性强弱分别将其分配到适合的操作系统上运行。ROS 操作系统负责管理建图、定位、路径规划、图像处理这类复杂程度较高和数据共享量较大的任务，FreeRTOS 则主要负责管理电机控制、冲撞检测这类及时性需求较高的任务。混合软件系统架构如图 1.5.3.4 所示。

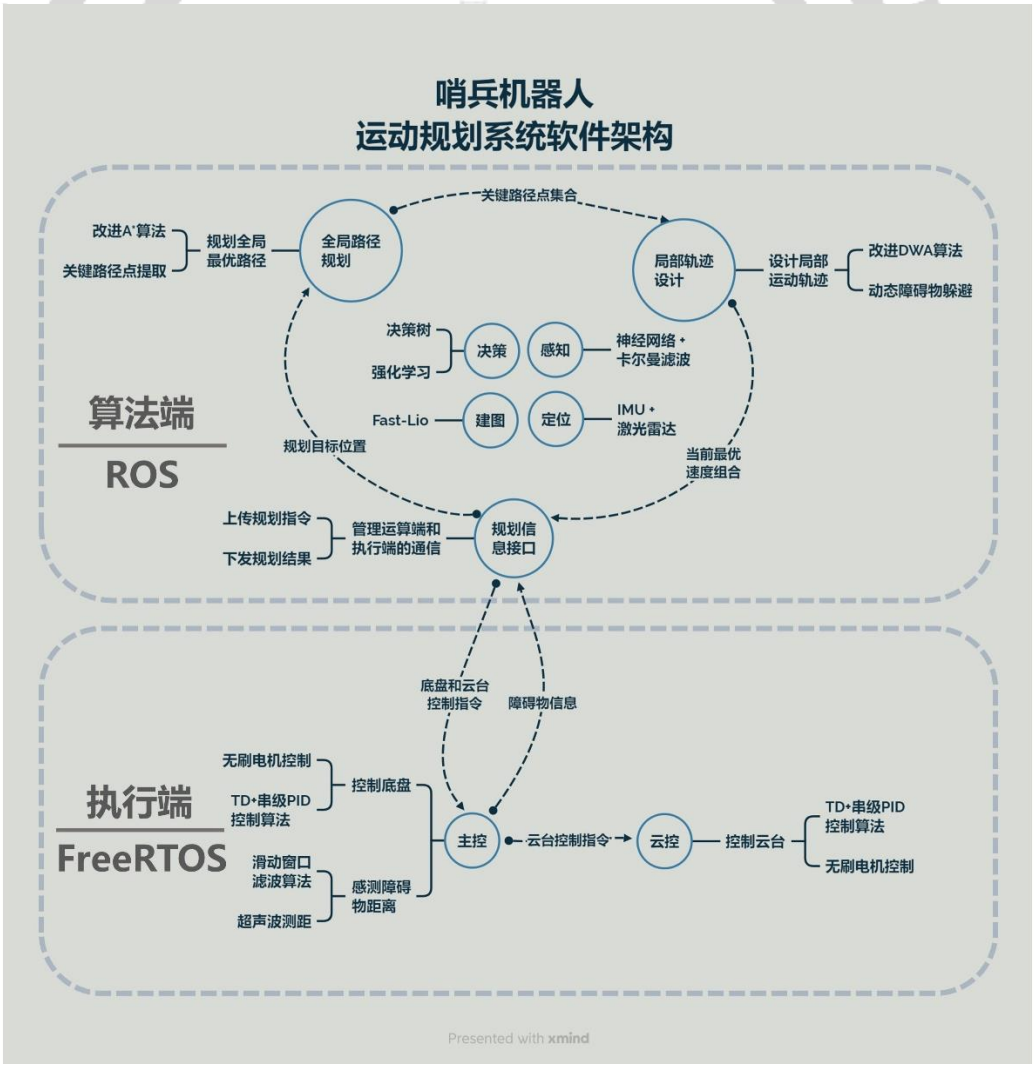


图 1.5.3.4　哨兵机器人混合软件架构

基于 ROS 的软件管理

ROS（Robot Operation System）是 Willow Garage 开源的机器人操作系统，它在 PC 层面为机器人程序开发提供了一些标准服务，使得机器人的开发与研究成果有了更强的复用性和移植性。另外，ROS 框架提供了一种基于话题发布/订阅的灵活的通讯机制，如图 1.5.3.5 所示，基于此机制，这让我们可以模块化地开发机器人的各项功能，然后利用这种通信机制为各个软件模块建立联系。比如本课题就利用它分别搭建了路径规划模块、定位模块、坐标变换模块和下位机通信模块并构建其完备的信息共享体系。同时，ROS 集成了许多其他开源项目的代码，因此可以使用多种语言进行编程，或者利用开源代码库来扩展机器人功能，诸如机器人控制、坐标系变换、可视化平台、物理仿真环境等。本课题利用坐标变换包来统一管理机器人包括雷达坐标系、相机坐标系、车身坐标系和世界坐标系等坐标系的变换，利用物理仿真环境和可视化平台来对所设计算法性能进行测试，利用机器人控制包来解算底盘各电机的实时速度。

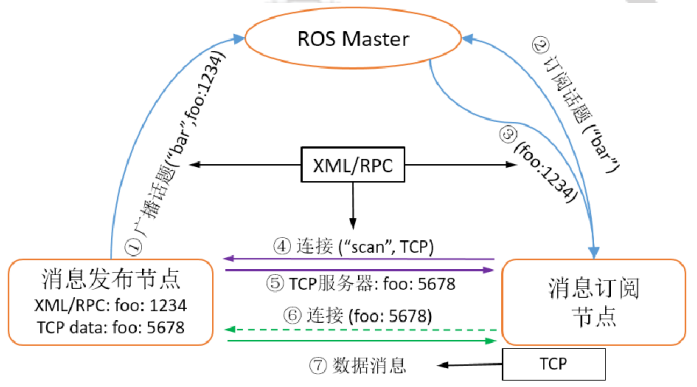


图 1.5.3.5　ROS 基础通信机制

基于 FreeRTOS 的实时控制

FreeRTOS 是一种轻量级的实时操作系统，非常适合部署在嵌入式设备上为其提供包括任务管理、时间管理、信号量和消息队列等功能。FreeRTOS 操作系统占用系统资源小而且调度灵活，可以方便地移植到各类单片机上，为实时性要求较高的任务提供方便的多任务管理系统，如图 1.5.3.6 所示。本课题利用 FreeRTOS 操作系统来管理电机驱动、冲撞检测和上位机通信等任务，为导航系统的执行端提供高效的控制服务。

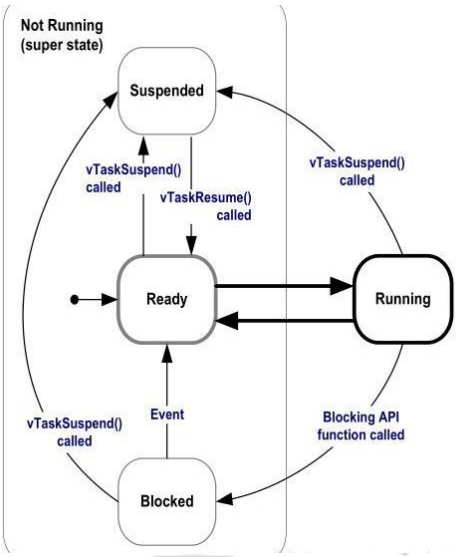


图 1.5.3.6　FreeRTOS 操作系统的多任务管理机制

1.5.4 算法设计

1.5.4.1 定位建图算法

定位作为哨兵机器人的基础模块之一，主要包括比赛前期赛场点云地图的建立和比赛时基于先验点云地图的定位两个部分。具体工作如下：

机器人定位需求分析

根据比赛要求，结合实际比赛中的场地信息和机器人的形态等特点，进行定位的精度和鲁棒性需求分析并提出合理的技术指标。

Gazebo 仿真环境搭建与传感器数据的获取

设计 Gazobo 仿真环境，通过 ROS 接口从中获取激光雷达和 IMU 的数据。搭建比赛场地及机器人的仿真模型，对机器人、场地环境以及传感器参数等进行设置，导出仿真环境中生成的数据。

基于激光雷达的点云地图构建

调研基于激光雷达的同时定位与建图算法。在仿真环境中利用多种类型的算法基于激光雷达和 IMU 数据建立比赛场地的点云地图，对不同算法实现的里程计精度和建图效果进行对比研究，确定满足要求的方法。

机器人定位算法设计

研究基于先验点云地图的移动机器人定位算法。利用事先建好的点云地图和实时的传感

器数据估计机器人相对于场地的位置信息。

算法实现与实验验证

对建图和定位算法进行仿真环境和实际场地的测试。在实际比赛中部署定位功能，设计合适的调试界面测试定位算法的精度与鲁棒性，判断是否满足指标要求，同时针对测试中的问题进行分析解决。

已完成的研究工作及成果

1.机器人定位需求分析

经过对比赛中定位任务的分析，提出了以下技术指标：

（1）定位算法运行稳定，不受其他移动机器人的影响；

（2）位置估计误差小于$10cm$；

（3）姿态（偏航角）估计误差小于$0.2rad$；

（4）定位算法帧率高于$10Hz$。

2.Gazebo 仿真环境搭建与传感器数据的获取

考虑到 Gazebo 与 ROS 具有丰富和完善的功能及接口，我们采用 Gazebo 搭建仿真环境，并用 ROS 获取机器人和传感器的数据。

2.1 基于 Gazebo 的仿真环境搭建

比赛场地的 3D 模型由官方提供。定位模块采用的传感器是激光雷达和 IMU，并不采集视觉相关的信息，所以未对赛场模型进行贴图，如图 1.5.4.1 所示。

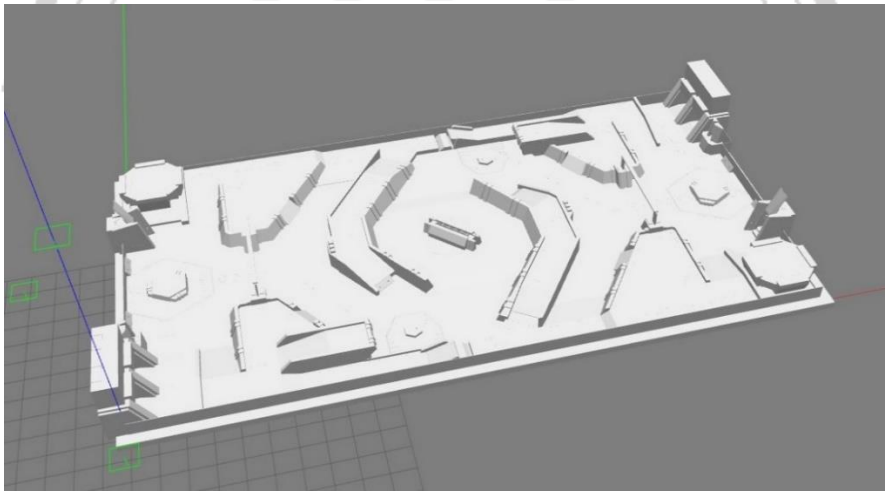


图 1.5.4.1　仿真环境中的比赛场地

在搭建好的仿真环境中，可添加用键盘控制移动的小车以及车载的激光雷达和 IMU 传感器，用于采集对应的传感器数据，如图 1.5.4.2 所示。这里我们采用的激光雷达模型是 Velodyne 32 线激光雷达 HDL-32E，IMU 模型是博世 BMI088。

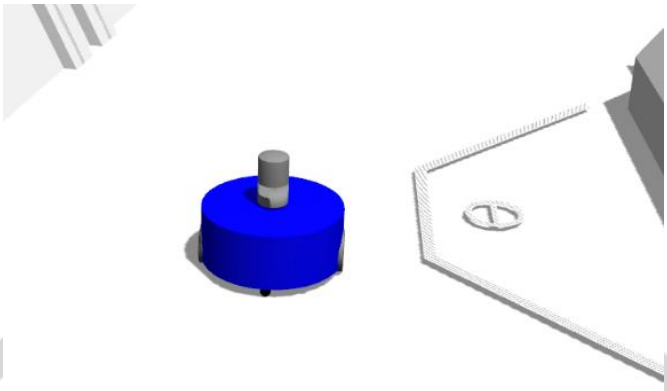


图 1.5.4.2　仿真环境中的移动小车和传感器

2.2 仿真数据的获取

在模型的配置文件中，可以设置 IMU 模型与激光雷达模型的平移和旋转关系，也就是传感器外参R_L^I和p_L^I。通过键盘输入可以改变小车的偏航角和运动速度，控制小车到达场地中的任意位置。在小车运动的过程中，我们使用 rosbag 工具将点云数据和 IMU 数据记录下来，还可以通过 Gazebo 读取小车模型的位姿真值用于里程计的精度评价。

3.基于激光雷达的点云地图构建

在机器人领域，同时定位与建图技术（Simultaneous Localization And Mapping，SLAM）在近年来取得了很大程度的进展。这类算法通过滤波或优化的方法对移动机器人的状态进行实时估计，并将机器人观测到的环境信息构建成为一致的地图。目前，基于激光雷达的 SLAM 形成了较为成熟的前后端的框架，前端基于点云配准进行实时定位，通常会用 ICP 及其变种或者 NDT 进行配准；后端包含回环检测和位姿图优化两部分。

在实际使用中，我们通常认为激光雷达采集的点云数据误差很小，可以通过里程计的精度评价构建点云地图的好坏。早期基于激光雷达的里程计为 LO（Lidar Odometry），通常是某种形式的 scan to scan 或者 scan to local map 配准，早期主要是基于 ICP、NDT 等的一些改良型配准算法。但纯 LO 的问题在于不能很好地去除点云畸变，并且存在诸多退化场景导致无法工作。后来出现了松耦合的 LIO（Lidar IMU Odometry），通过融合 IMU 的数据实现点云去畸变，IMU 观测也可以为点云配准提供初值，最终结果可以直接由点云配准得到，也可以用滤波融合得到。相关的研究成果有 LOAM (with IMU)，LeGO-LOAM 等。但松耦合 LIO

对 Lidar 和 IMU 的融合仅体现在结果层面，没有考虑两种观测间的内在约束，且依然无法解决一些退化场景问题。近年来又涌现了许多紧耦合的 LIO，这些工作主要分为迭代扩展卡尔曼滤波和滑窗优化两种状态估计框架，充分考虑了 Lidar 观测和 IMU 观测的内在约束性，互相影响，共同决定最终的结果。代表有 LINS、FAST-LIO、LIO-Mapping、LIO-SAM 等。

目前我们在仿真环境中部署了 FAST-LIO 的工作，并计算了他的位置估计误差和偏航角估计误差，结果如表 1.5.4.1 所示。可以看出该算法满足前述的指标要求。

表 1.5.4.1 仿真环境中 FAST-LIO 的里程计精度

仿真次数	偏航角最大误差/rad	偏航角 RMSE/rad	位置距离最大误差/cm	位置距离 RMSE/cm
1	0.51	0.05	7. 93	1.43
2	0.20	0.03	5.26	1.16
3	0.37	0.02	8.76	1.24
4	0.28	0.02	4.31	0.75
5	0.14	0.01	4.60	0.76

后续计划是在仿真中将 state of the art 算法都测试一下，进行横向对比。因为到时候比赛官方会单独给建图时间，所以不需要考虑建图算法的运算速度或建图时哨兵的机动性能。只需要让车运动得慢一些、平缓一些，然后分别计算各个算法的位姿估计误差，选出建图过程中里程计精度最高的进行使用。

4．机器人定位算法设计

我们选用的 IMU 传感器获取数据的帧率为 100Hz，激光雷达获取点云数据的帧率为 10Hz，所以在每次点云数据到来之前，先利用 IMU 的数据进行机器人的状态递推，在接收到点云数据之后，再利用递推得到的状态（旋转和平移）对点云进行畸变去除，然后将去除畸变后的点云与地图进行匹配，最后进行 IEKF 的更新。整个系统的框架如图 2-3 所示，其中初始化部分确定了机器人在地图中的初始位姿、陀螺仪和加速度计测量偏差的初始值、重力在第一帧 IMU 坐标系下的表示。

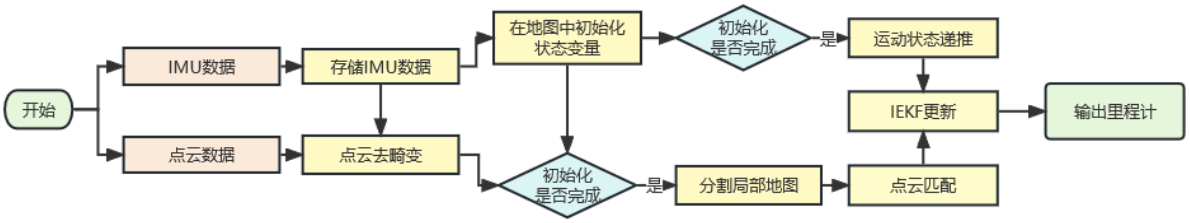


图 1.5.4.3 定位算法框架

4.1 坐标系定义

定位系统一共涉及的四个坐标系，如图 1.5.4.4 所示。其中世界坐标系静止，规定为和第一帧 IMU 坐标系重合；机器人坐标系规定为和当前帧的 IMU 坐标系重合；激光雷达坐标系相对当前的 IMU 坐标系具有一定的平移和旋转关系，由机械安装确定。

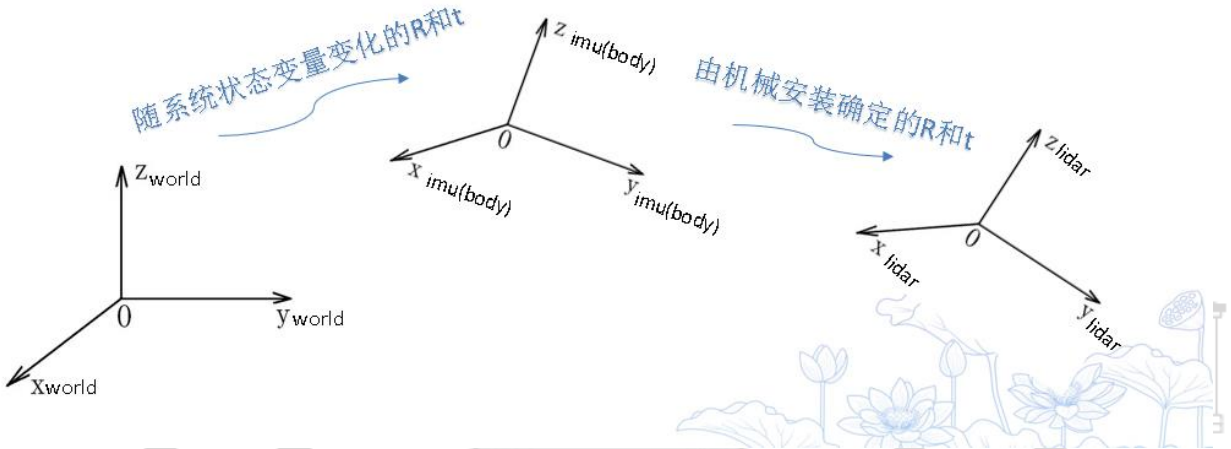


图 1.5.4.4 定位系统涉及的坐标系

4.2 IMU 状态递推模型

系统维护的状态变量包含机器人在世界坐标系中的位置p^G、机器人在世界坐标系中的速度v^G、机器人坐标系相对世界坐标系的旋转R^G、陀螺仪的测量偏差b^ω、加速度计的测量偏差b^a、激光雷达坐标系相对 IMU 坐标系的平移p_L^I和旋转R_L^I、重力在世界坐标系下的表示g^G。

根据匀加速度和匀角速度运动模型，连续时间下的状态递推模型为：

$$\dot{p}^G = v^G \tag{1.5.4-1}$$

$$\dot{v}^G = R^G(a - b^a - n^a) + g^G \tag{1.5.4-2}$$

$$\dot{R}^G = R^G[\omega - b^\omega - n^\omega]_\times \tag{1.5.4-3}$$

$$\dot{b}^\omega = n^\omega \tag{1.5.4-4}$$

$$\dot{b}^a = n^a \tag{1.5.4-5}$$

$$\dot{R}_L^I = 0 \tag{1.5.4-6}$$

$$\dot{p}_L^I = 0 \tag{1.5.4-7}$$

$$\dot{g}^G = 0 \tag{1.5.4-8}$$

其中a为加速度计的测量值，n^a为加速度计的测量噪声（假设为零均值白噪声，满足正态分布，即$n^a \sim N(0, \sigma_{n^a}^2)$），$\omega$为陀螺仪的测量值，$n^\omega$为陀螺仪的测量噪声（假设为零均值白噪声，满足正态分布，即$n^\omega \sim N(0, \sigma_{n^\omega}^2)$），$[\cdot]_\times$为中括弧中向量对应的反对称矩阵。

将连续状态模型推广到离散状态，得到：

$$p_{i+1}^G = p_i^G + \Delta t \cdot v_i^G + \frac{1}{2}\Delta t^2 \cdot \left[R_i^G(a_{i+1} - b_i^a - n^a) + g^G\right] \tag{1.5.4-9}$$

$$v_{i+1}^G = v_i^G + \Delta t \cdot \left[R_i^G(a_{i+1} - b_i^a - n^a) + g^G\right] \tag{1.5.4-10}$$

$$R_{i+1}^G = R_i^G \cdot Exp[\Delta t \cdot (\omega_{i+1} - b_i^\omega - n^\omega)] \tag{1.5.4-11}$$

$$b_{i+1}^\omega = b_i^\omega + n^\omega \tag{1.5.4-12}$$

$$b_{i+1}^a = b_i^a + n^a \tag{1.5.4-13}$$

$$R_L^I = R_L^I \tag{1.5.4-14}$$

$$p_L^I = p_L^I \tag{1.5.4-15}$$

$$g^G = g^G \tag{1.5.4-16}$$

其中Δt为第$i+1$时刻和第i时刻的时间差，$Exp[\cdot]$代表从欧拉角到旋转矩阵的变换。

根据状态递推公式(2-9)~(2-16)，我们可以得到第i时刻（对应每一帧 IMU 数据的时间戳）系统的先验状态变量$\check{x}_i = \left[\check{p}_i^G,\ \check{v}_i^G,\ \check{R}_i^G,\ \check{b}_i^\omega,\ \check{b}_i^a,\ \check{p}_L^I,\ \check{R}_L^I,\ \check{g}^G\right]^T$。此外，我们采用误差状态进行计算，原因如下：误差状态自由度与姿态自由度相等，避免过度参数化；两帧雷达观测的时间间隔很短，所以误差状态在 0 附近，线性化操作更加准确；误差状态是小量，二阶项可以直接忽略，方便雅可比计算。

算法经过一次 IEKF 更新之后，在下一帧点云数据到来之前，利用两帧点云数据之间的 IMU 数据对误差状态进行递推。在点云数据到来时，根据观测修正当前的误差状态，并将其添加到系统本身的状态变量上。然后将误差状态置零，重复上述步骤，在下一帧点云数据到来前继续对误差状态进行递推。假设第i时刻 IMU 递推得到的先验误差状态为$\delta\check{x}_i = \left[\delta\check{p}_i^G,\ \delta\check{v}_i^G,\ \delta\check{\theta}_i^G, \delta\check{b}_i^\omega, \delta\check{b}_i^a, \delta\check{p}_L^I,\ \delta\check{R}_L^I,\ \delta\check{g}^G\right]^T$，该误差状态对应的协方差为$\check{P}_i$，第$k$时刻（在第$i$时刻之前且与第$i$时刻最相近的雷达数据采集时间）经过 IEKF 融合更新的姿态后验估计为$\hat{R}_k^G$，IMU 测量偏差的后验估计为$\hat{b}_k^a$和$\hat{b}_k^\omega$，则整个系统的先验误差状态及其协方差的递推过程如下：

$$\delta\check{p}_{i+1}^G = \delta\check{p}_i^G + \delta\check{v}_i^G \cdot \Delta t \tag{1.5.4-17}$$

$$\delta\check{v}_{i+1}^{G}=\delta\check{v}_{i}^{G}+\left\{-\hat{R}_{k}^{G}\left[a_{i+1}-\hat{b}_{k}^{a}\right]_{\times}\delta\check{\theta}_{i}^{G}-\hat{R}_{k}^{G}\delta\check{b}_{i}^{a}+\delta\check{g}^{G}\right\}\cdot\Delta t \tag{1.5.4-18}$$

$$\delta\check{R}_{i+1}^{G}=\hat{R}_{k}^{G\,T}\left[\left(\omega_{i+1}-\hat{b}_{k}^{\omega}\right)\Delta t\right]\delta\check{\theta}_{i}^{G}-\delta\check{b}_{i}^{\omega}\cdot\Delta t \tag{1.5.4-19}$$

$$\delta\check{b}_{i+1}^{\omega}=\delta\check{b}_{i}^{\omega} \tag{1.5.4-20}$$

$$\delta\check{b}_{i+1}^{a}=\delta\check{b}_{i}^{a} \tag{1.5.4-21}$$

$$\delta\check{R}_{L}^{I}=\delta\check{R}_{L}^{I} \tag{1.5.4-22}$$

$$\delta\check{p}_{L}^{I}=\delta\check{p}_{L}^{I} \tag{1.5.4-23}$$

$$\delta\check{g}^{G}=\delta\check{g}^{G} \tag{1.5.4-24}$$

$$\check{P}_{i+1}=F_{x}\check{P}_{i}F_{x}^{T}+F_{w}Q_{i}F_{w}^{T} \tag{1.5.4-25}$$

其中F_x为状态递推方程相对误差状态的雅可比矩阵，F_w为状态递推方程相对 IMU 测量值的雅可比矩阵，Q_i为 IMU 测量值对应的协方差，它们的具体形式如下：

$$F_{x}=\begin{bmatrix}I & I\cdot\Delta t & 0 & 0 & 0 & 0 & 0\\ 0 & I & -\hat{R}_{k}^{G}\left[a_{i+1}-\hat{b}_{k}^{a}\right]_{\times}\cdot\Delta t & -\hat{R}_{k}^{G}\cdot\Delta t & 0 & 0 & I\cdot\Delta t\\ 0 & 0 & \hat{R}_{k}^{G\,T}\left[\left(\omega_{i+1}-\hat{b}_{k}^{\omega}\right)\Delta t\right] & 0 & -I\cdot\Delta t & 0 & 0\\ 0 & 0 & 0 & I & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & I & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & I & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & I\\ 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \tag{1.5.4-26}$$

$$F_{w}=\begin{bmatrix}0 & 0 & 0 & 0\\ I & 0 & 0 & 0\\ 0 & I & 0 & 0\\ 0 & 0 & I & 0\\ 0 & 0 & 0 & I\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \tag{1.5.4-27}$$

$$Q_{i}=\begin{bmatrix}\sigma_{n^{\omega}}^{2}\cdot\Delta t^{2}\cdot I & 0 & 0 & 0\\ 0 & \sigma_{n^{a}}^{2}\cdot\Delta t^{2}\cdot I & 0 & 0\\ 0 & 0 & \sigma_{n^{w_{\omega}}}^{2}\cdot\Delta t^{2}\cdot I & 0\\ 0 & 0 & 0 & \sigma_{n^{w_{a}}}^{2}\cdot\Delta t^{2}\cdot I\end{bmatrix} \tag{1.5.4-28}$$

其中$\sigma_{n^{w_{\omega}}}^{2}$和$\sigma_{n^{w_{a}}}^{2}$为陀螺仪和加速度计随机游走的方差。

4.3 点云畸变的去除

由于激光雷达完成一帧点云的扫描需要一定时间，而在这段时间内机器人会发生运动，导致获取到的点云数据缺乏一致性，被称为点云畸变。在实际应用中，我们需要对点云进行去畸变处理。首先确定每一帧点云的第一个点采集的时刻t_k，以及这一帧中每个点ρ_j采集的时刻j所在的 IMU 数据时间区间$[\tau_{i-1},\ \tau_i]$，然后根据存储的系统状态变量和传感器测量数据将第j时刻的点从该时刻的激光雷达坐标系中变换到t_k时刻的激光雷达坐标系中。

各个时刻点的坐标变换关系如下：

$$R_L^I \rho_j^L + p_L^I = \rho_j^I \tag{1.5.4-29}$$

$$R_j^G \rho_j^I + p_j^G = \rho^G \tag{1.5.4-30}$$

$$R_L^I \rho_{t_k}^L + p_L^I = \rho_{t_k}^I \tag{1.5.4-31}$$

$$R_{t_k}^G \rho_{t_k}^I + p_{t_k}^G = \rho^G \tag{1.5.4-32}$$

其中$\cdot^L$表示在激光雷达坐标系下，$\cdot^I$表示在 IMU 坐标系下，$\cdot^G$表示在世界坐标系下。ρ^G为需要去除畸变的点在世界坐标系中的坐标，是恒定不变的；ρ_j^L为该点在第j时刻激光雷达坐标系中的坐标，由激光雷达测量，是已知量；R_j^G和p_j^G为第j时刻机器人坐标系和世界坐标系的旋转和平移关系，由存储的状态变量和 IMU 测量求得，求解过程为：

$$R_{\rho_j}^G = \hat{R}_{i-1}^G \cdot Exp(\omega_{i-1} \cdot dt) \tag{1.5.4-33}$$

$$p_{\rho_j}^G = \hat{p}_{i-1}^G + \hat{v}_{i-1}^G \cdot dt + \frac{1}{2} a_{i-1} \cdot dt^2 \tag{1.5.4-34}$$

其中dt为第j时刻和τ_{i-1}时刻的时间差，$\hat{R}_{i-1}^G$、$\hat{p}_{i-1}^G$、$\hat{v}_{i-1}^G$为τ_{i-1}时刻系统的后验状态估计，ω_{i-1}、a_{i-1}为τ_{i-1}时刻 IMU 的测量值。

联立公式(1.5.4-29)~(1.5.4-34)，即可求得t_k时刻的激光雷达坐标系中该点的坐标$\rho_{t_k}^L$。对每一帧点云的每一个点都进行上述操作，便使得每个点都处于所在帧的点云中第一个点采集时刻的激光雷达坐标系中，也就去除了运动畸变。

4.4 激光雷达观测更新

在 LIO 定位系统中，激光雷达的观测数据以较低的频率到来，获取的点云数据去除畸变后，需要融合到前述 IMU 的递推状态上，以修正 IMU 的累计漂移误差，主要分为以下三步：

(1) 以点到面的模型将当前帧的点与地图匹配

设第k时刻雷达中去畸变之后的点集合为$\{\rho_j^L,\ j = 1, 2, \dots, m\}$，其中$m$为每一帧点云中点的个数。对于每一个点$\rho_j^L$，根据与第$k$时刻最接近的 IMU 先验状态估计（假设为第$i$时刻）将其变换到世界坐标系下：

$$\rho_j^G = \check{R}_i^G \left(R_L^I \rho_j^L + p_L^I\right) + \check{p}_i^G \tag{1.5.4-35}$$

此时ρ_j^G和历史帧累积构成的地图均处于世界坐标系下，可以在地图中使用最邻近搜索得到五个和p_j^G距离最近的点（为加速搜索，这里的地图使用 kd 树结构存储）。我们使用这五个点拟合平面，得到平面的单位法向量u_j^G，设q_j^G为平面上的相关点，如图 1.5.4.5 所示。

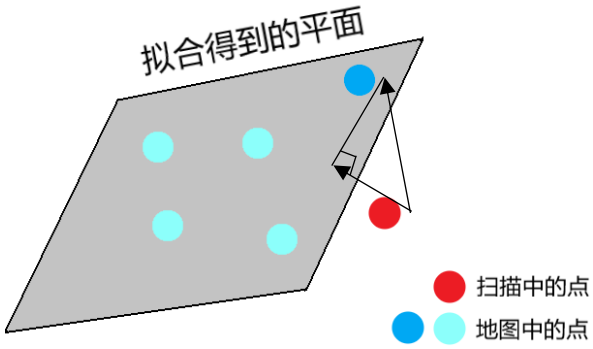


图 1.5.4.5 点到面的匹配示意图

(2) 考虑激光雷达的测量噪声和先验状态的误差构建残差模型

假设对于点ρ_j^L激光雷达的测量噪声为n_j^L，则测量点的真值为$\bar{\rho}_j^L = \rho_j^L + n_j^L$。将测量点的真值变换到世界坐标系下，并求出它和匹配平面的距离d：

$$d = u_j^{G^T}\left[\check{R}_i^G\left(R_L^I\bar{\rho}_j^L + p_L^I\right) + \check{p}_i^G - q_j^G\right] \tag{1.5.4-36}$$

如果先验状态没有误差，则测量点到关联平面的距离为 0，即$d = 0$。但是 IMU 的状态递推存在一定误差，测量估计的目的就是调整状态变量的值使距离残差d最小化。

(3) IEKF 融合更新

根据式(1.5.4-36)定义目标函数：

$$h_j\left(\check{x}_i + \delta x_i, \rho_j^L + n_j^L\right) = u_j^{G^T}\left[\check{R}_i^G\left(R_L^I\bar{\rho}_j^L + p_L^I\right) + \check{p}_i^G - q_j^G\right] = 0 \tag{1.5.4-37}$$

对目标函数进行一阶近似：

$$h_j\left(\check{x}_i + \delta x_i, \rho_j^L + n_j^L\right) \cong h_j\left(\check{x}_i, \rho_j^L\right) + H_j \cdot \delta x_i + v_j \tag{1.5.4-38}$$

其中$h_j\left(\check{x}_i, \rho_j^L\right) = u_j^{G^T}\left[\check{R}_i^G\left(R_L^I\rho_j^L + p_L^I\right) + \check{p}_i^G - q_j^G\right]$。假设$x_i$为系统在第$i$时刻状态变量的真实值，则$\delta x_i$为真实状态和先验状态之间的差值，即$\delta x_i = x_i - \check{x}_i$。$H_j$为真实状态下残差$h_j\left(\check{x}_i + \delta x_i, \rho_j^L + n_j^L\right)$相对$\delta x_i$的雅可比矩阵。$v_j$来源于激光雷达的测量噪声$n_j^L$，满足零均值正态分布$v_j \sim N\left(0, R_j\right)$。

设在第α次迭代时的先验状态变量为x_i^α（$\alpha = 0$时有$x_i^\alpha = \check{x}_i$）。定义残差：

$$r_i^\alpha = 0 - h_j\left(x_i^\alpha, \rho_j^L\right) = h_j\left(x_i^\alpha + \delta x_i^\alpha, \rho_j^L + n_j^L\right) - h_j\left(x_i^\alpha, \rho_j^L\right) \tag{1.5.4-39}$$

其中δx_i^α为x_i^α和真实状态x_i之间的差值。根据一阶近似的结果，式(2-39)可写为：

$$r_i^\alpha \cong H_j^\alpha \cdot \delta x_i^\alpha + D_j^\alpha \cdot v_j \tag{1.5.4-40}$$

其中H_j^α为真实状态下残差$h_j\left(x_i^\alpha+\delta x_i^\alpha, p_j^L+n_j^L\right)$相对$\delta x_i^\alpha$的雅可比矩阵，$D_j^\alpha$为真实状态下残差$h_j\left(x_i^\alpha+\delta x_i^\alpha, p_j^L+n_j^L\right)$相对$n_j^L$的雅可比矩阵，因此误差状态在第$\alpha$次迭代时的观测分布为：

$$r_i^\alpha - H_j^\alpha \cdot \delta x_i^\alpha \sim N\left(0, R_j^\alpha\right) \tag{1.5.4-41}$$

其中$R_j^\alpha = D_j^\alpha \cdot R_j \cdot D_j^{\alpha T}$。

同时，根据 IMU 的状态递推结果我们可以得到第i时刻系统误差状态的先验分布为$\widetilde{\delta x_i} = x_i - \check{x}_i \sim N\left(0, \check{P}_i\right)$。在第$\alpha$次迭代时$\widetilde{\delta x_i^\alpha} = x_i - x_i^\alpha$，将$\widetilde{\delta x_i^\alpha}$在$\widetilde{\delta x_i}$处一阶展开：

$$\widetilde{\delta x_i^\alpha} \cong x_i^\alpha - \check{x}_i + \left(J_i^\alpha\right)^{-1} \cdot \delta x_i^\alpha \tag{1.5.4-42}$$

其中J_i^α为真实的误差状态$\delta x_i^\alpha = [\check{x}_i + (x_i^\alpha - \check{x}_i)] - x_i^\alpha$相对 $(x_i^\alpha - \check{x}_i)$的雅可比矩阵。则误差状态在第$\alpha$次迭代时的先验分布为：

$$\widetilde{\delta x_i^\alpha} \sim N(-J_i^\alpha (x_i^\alpha - \check{x}_i), P_i^\alpha) \tag{1.5.4-43}$$

其中$P_i^\alpha = J_i^\alpha \cdot \check{P}_i \cdot J_i^{\alpha T}$。

对式(2-41)和式(2-43)的分布做最大后验估计：

$$\widehat{\delta x_i^\alpha} = \underset{\delta x_i^\alpha}{argmin}\left(\left\|r_i^\alpha - H_j^\alpha \cdot \delta x_i^\alpha\right\|^2_{R_j^{\alpha -\frac{1}{2}}} + \left\|x_i^\alpha - \check{x}_i + \left(J_i^\alpha\right)^{-1} \cdot \delta x_i^\alpha\right\|^2_{P_i^{\alpha -\frac{1}{2}}}\right) \tag{1.5.4-44}$$

其中$\|x\|_A^2 = \|Ax\|^2 = x^T A^T A x$。重复上述步骤直到$\left\|\widehat{\delta x_i^\alpha} - \widehat{\delta x_i^{\alpha-1}}\right\| < \epsilon$（$\epsilon$为设置的极小值）或迭代次数$\alpha$足够大，则迭代结束，得到误差状态的最终后验估计$\widehat{\delta x_i} = \widehat{\delta x_i^\alpha}$。 将最终的误差状态$\widehat{\delta x_i}$加到递推得到的先验状态$\check{x}_i$上，便是系统的后验状态$\hat{x}_i$。后验状态中的旋转和平移就是我们需要的里程计信息。

4.5 机器人位姿的初始化

机器人在先验地图中的初始位姿可以通过测量或估算等手段获取，但是精度有所欠缺，所以在粗略估计的基础上还需要在静止情况下将激光雷达的当前帧点云与粗略位置周围的地图点云进行匹配，得到较为精确的初始位姿。点云匹配的手段通常有 NDT、ICP 及其变种。NDT 方法将已知点云空间划分为栅格，计算各栅格均值、协方差，根据预测的位姿计算联合概率，最大化联合概率，优化求解位姿。ICP 方法分为点到点、点到直线、点到平面等不同的关联模型，均先通过预测的位姿关联点和点（直线/平面），然后最小化所有点到点（直线/平面）的距离之和，优化求解位姿。

我们实现了点到点的 ICP 和点到平面的 ICP 初始化，用高斯牛顿法进行优化，发现 ICP 算法对偏航角的初值非常敏感，如果偏航角差了 15° 以上基本就会初始化失败。所以修改为

水平 360° 方向每隔 20° 计算一次 ICP，取优化结果误差最小的偏航角作为初始化结果。经过修改后测试初始化过程非常稳定，但是因为要计算 18 次 ICP，虽然进行了并行化处理，花费的时间仍然较长，约 20-30 秒，所以在运算速度方面还需要进一步优化。

此外，由于 ICP 是点与点之间的匹配，NDT 则计算点云的分布信息，NDT 在一般情况下比 ICP 更加鲁棒。在位姿初始化步骤中最重要的就是稳定性和鲁棒性，所以我们考虑后续实现 NDT 算法进行初始化，与 ICP 算法进行比较。

4.6 局部地图的分割

在位姿初始化之前，需要根据先验位置进行局部地图的分割；在初始化之后将激光扫描和地图匹配时，也需要根据前一时刻的先验位置分割地图。为了保证点云匹配的速度，局部地图不能过大；但如果局部地图过小，将导致频繁的分割操作，降低程序的运行效率。

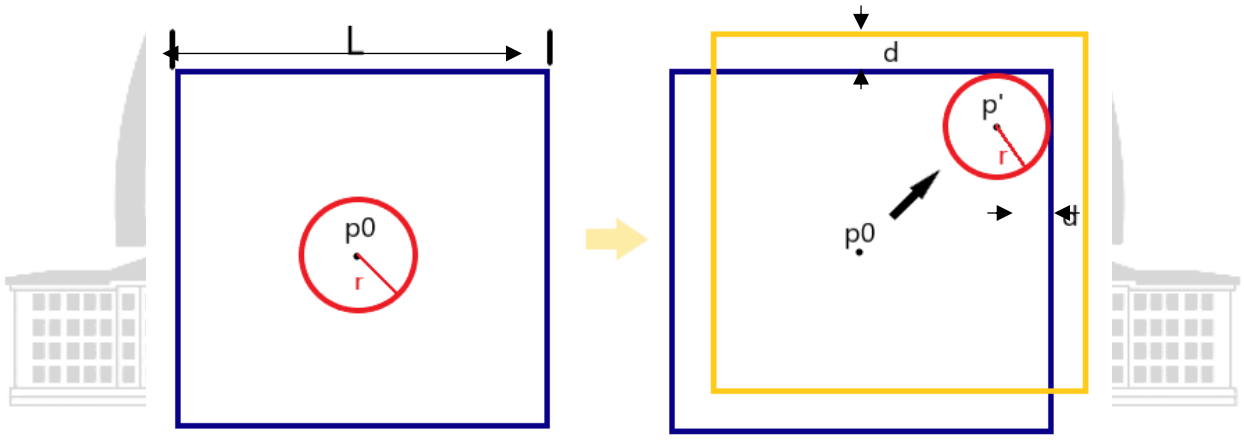


图 1.5.4.6　局部地图分割示意图

如图 1.5.4.6 所示，首先根据初始位姿p_0分割边长为L的立方体（图片为三维地图和扫描在二维平面上的投影）。当激光雷达从p_0运动到p'时，以p'为中心扫描半径为r的圆触碰到了局部地图的边界，此时我们将局部地图范围向被触碰的方向移动d的距离，一般取d为扫描半径r的倍数。图中黄色边框就是原始局部地图范围（蓝色边框）移动的结果，得到新的范围后程序将对点云地图重新进行分割。

5 算法实现与测试

由于 ROS 具有丰富的传感器驱动程序、较为完善的通讯协议和便利的调试工具，我们采用 C++在 ROS 框架下实现所有的算法并进行测试。

5.1 停车场数据集测试

我们使用一个在停车场中通过 Livox Avia 采集的点云和 IMU 数据集以及该停车场的地图来测试定位算法，数据集中激光雷达的数据的帧率为 10Hz，IMU 数据的帧率为 200Hz，时间跨度为 800s。因为停车场和比赛场地都是高度结构化的室内环境且存在一定对称性，所以能够较好地验证算法的可行性。

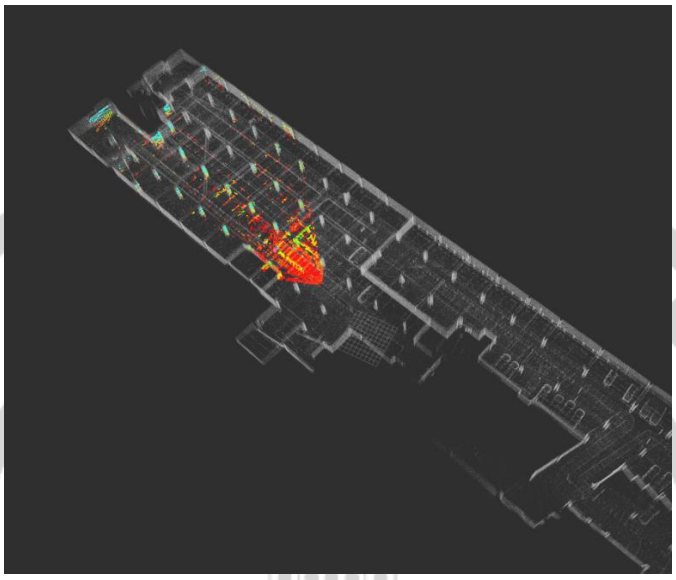


图 1.5.4.7　基于停车场点云地图的位姿初始化

在给定数据集的粗略位姿后，进行位姿的初始化。初始化的结果如图 1.5.4.7 所示，可以看出当前帧的点云（彩色点）与地图（白色点）具有很好的重合度，代表位姿初始化成功。初始化完成之后，根据 4 中的算法框架进行实时的定位并输出里程计。定位的结果如图 1.5.4.8 所示，其中绿色曲线为机器人运动的轨迹，白色点为先验地图。可以看出在运行过程中定位算法运行稳定且轨迹非常光滑，当前帧点云与地图的重合度良好，证明了定位算法的可行性。

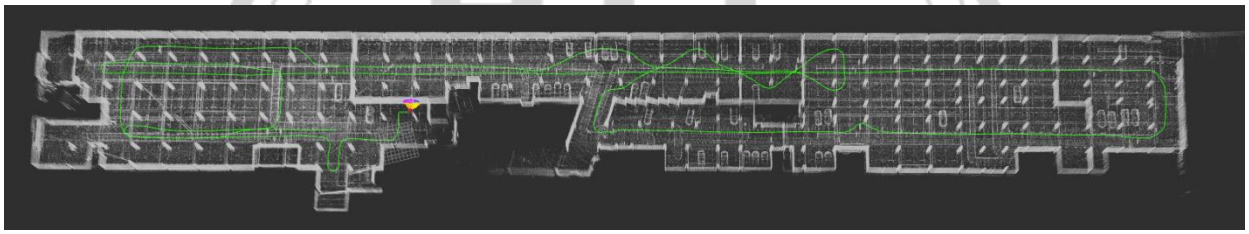


图 1.5.4.8 基于停车场点云地图的定位效果

5.2 仿真环境下测试与里程计精度评价

我们在 2 中搭建的仿真环境下通过键盘指令控制小车移动，利用产生的数据测试了建图和定位算法，如图 2-9 所示。

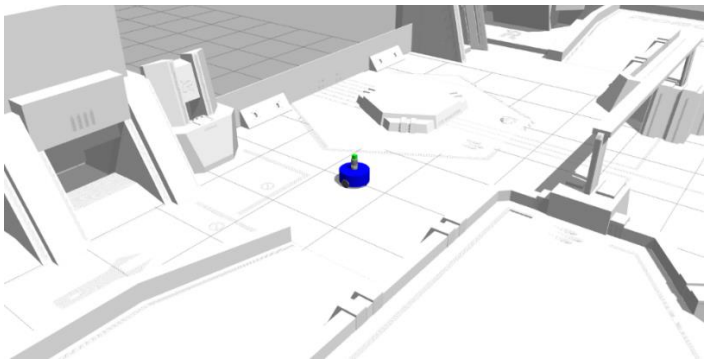


图 1.5.4.9 控制小车在比赛场里中移动采集数据

小车经过指定位置后，将建好的点云地图保存供定位算法使用，如图 1.5.4.10 所示。

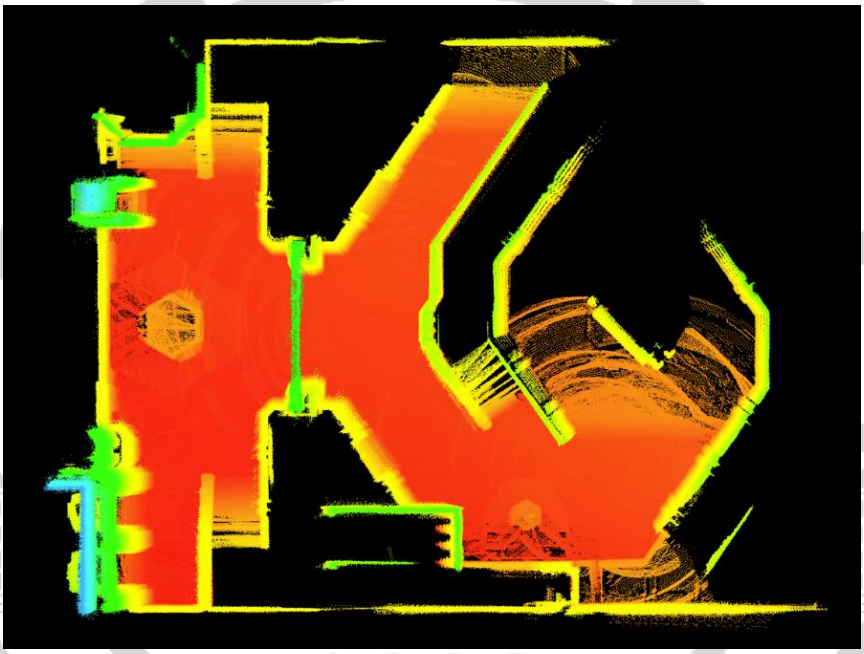


图 1.5.4.10 左半部分比赛场地的点云地图（俯视）

重新启动仿真控制小车移动测试定位算法，如图 1.5.4.11 所示。可以看出扫描得到的点云与场地的点云地图具有良好的匹配效果。此外，我们同时记录了里程计的输出和通过 Gazebo 获取的位姿真值，利用位置信息将两段轨迹对齐后对里程计的精度进行了评价。我们进行了多次仿真并对每次记录的数据分别计算了偏航角误差的最大值、偏航角的均方根误差、位置距离误差的最大值以及位置距离的均方根误差，所有的评价结果总结在表 1.5.4.2 中。

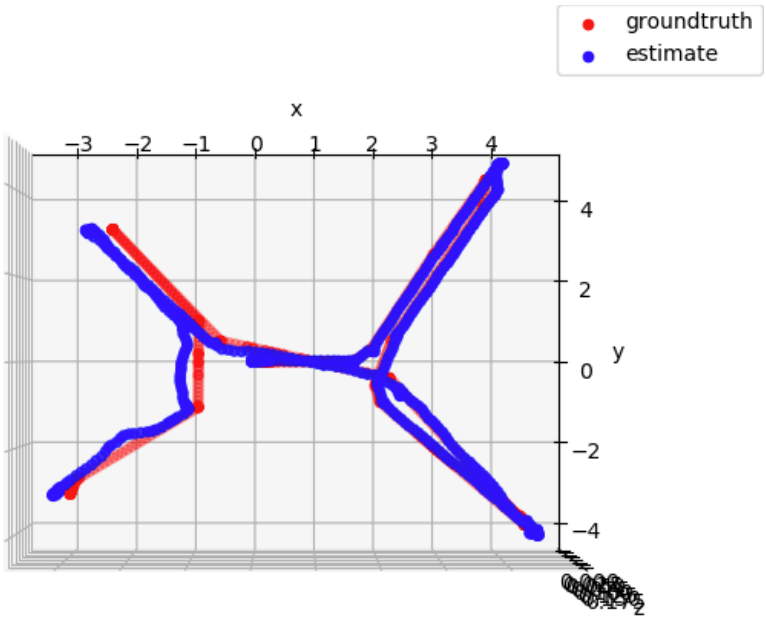


图 1.5.4.11 仿真环境下的定位效果（所有坐标单位为 m）

在测试中我们发现偏航角误差超出了预期，排查原因发现小车在进行较大的偏航机动时 IMU 递推的先验状态误差较大。所以限制了陀螺仪和加速度计输出的最大值，再次进行了测试，最终的评价结果满足设计的指标要求。另外在哨兵机器人的结构上设计了二级云台，控制雷达的偏航角一直朝着固定方向，这样偏航角的误差就更小了。

表 1.5.4.2 仿真环境中里程计精度评价结果

仿真次数	偏航角最大误差/rad	偏航角 RMSE/rad	位置距离最大误差/cm	位置距离 RMSE/cm
1	0.23	0.02	4.73	1.14
2	0.16	0.02	5.09	1.20
3	0.28	0.03	4.31	0.75
4	0.19	0.02	4.12	0.90
5	0.14	0.01	4.60	0.76

1.5.4.2 运动规划算法

1.功能简介

运动规划算法直接决定了哨兵机器人在 RM 赛场上的机动性，而机动性正是竞技机器人最重要的属性之一。为了让哨兵机器人具备在复杂动态场景下安全高效的移动能力，我们设计了一套优化 A*-DWA 运动规划算法。本算法能够自主规划出从当前位置到任意指定赛场位

置的全局移动路径，再结合本地传感器（激光雷达、视觉系统、超声波测距等）捕获的实时局部赛场信息，获取满足避障需求的局部移动轨迹，最后实时更新符合机器人运动学和动力学要求的移动速度以驱动底盘实现自主移动。

2.算法原理与公式推导

2.1 全局路径规划

全局路径规划用于设计从当前位置到目标位置的最优移动路径。哨兵机器人的全局规划以 A*算法为基础，结合碰撞约束和机器人运动学对 A*规划结果进行两次路径优化设计，最终得到一条安全高效的全局路径。全局规划算法原理如下：

（1）利用激光雷达采集的三维全局点云地图降采样得到二维栅格地图。

（2）根据碰撞约束对栅格地图进行膨胀处理（膨胀尺度以车身外接圆半径大小为参考）。

（3）执行 A*算法获取从当前位置到目标位置的全局路径。

由于 A*是一种实时最优算法，实时最优并不等于全局最优，可能会有局部不可知的障碍物阻挡在实时最优方向上，从而出现移动机器人走错路回头的情况，难以避免会产生冗余路段。为了消除这些冗余路段，我们设计了一种路径关键点提取算法，经过该算法处理后的全局路径更加简短，拐角数目更少且转弯角度更小。第一次优化首先获取 A*路径中的全部路径节点$\{x_1, x_2, \dots x_{n-1}, x_n\}$，其中$x_1$为起点，$x_n$为目标节点，$V$为优化后的关键点集合。以$x_1$为出发点，$m$从 3 开始增加，依次直线连接$x_1$和$x_m$，如果$x_1$和$x_m$的连线经过障碍物，说明$x_m$是一个转折点，将$x_{m-1}$加入关键点集合$V$。把$x_{m-1}$设置为新出发点，继续增加$m$，重复转折点查找过程，直到$m = n$，把目标节点$x_n$加入$V$，完成关键点提取算法。

A*算法在寻找最短路径时，会产生贴着障碍物边缘的路径，但是栅格地图具有障碍物边缘粗糙的特点，经过第一次路径优化后仍会将一系列伪关键点加入关键点集合，从而导致在贴障路段存在相邻路径点间距离太近、转角太大的现象，这种高频的路径点突变会对后续的轨迹规划和机器人移动造成状态突变的影响。为了解决这个问题，我们又在第一次路径优化的结果上应用了第二次路径优化，经过第二次优化后的全局路径更加平滑，路径节点更少。第二次优化首先检测所有贴障路径点的周围空间的障碍物，然后将贴障路径点往空旷区域移动一段距离，再检测移动后的相邻全局路径点之间是否存在障碍物，如果存在障碍物则在相邻路径点中间生成新的路径点，最后对修补后的移动全局路径进行第二次关键点提取，得到最终的全局规划路径结果。

检测到临时障碍物时执行重规划。

2.2 局部轨迹规划

局部轨迹规划用于生成移动机器人实时移动轨迹和设计实时移动速度。哨兵机器人的轨迹规划算法以 DWA 算法为基础，结合碰撞约束、运动学约束和动力学约束设计满足实时避障需求的移动轨迹，使哨兵机器人能够灵活躲避先验障碍物、临时障碍物和移动障碍物。局部轨迹规划的原理如下：

（1）利用实时局部障碍物信息、运动学约束和动力学约束设计当前运动周期下的允许线速度范围$[v_{min}, v_{max}]$和允许角速度$[\omega_{min}, \omega_{max}]$范围。

a) 运动学约束

移动机器人自身最大最小速度限制为：

$$V_s = \{(v, \omega), v \in [v_{\min}, v_{\max}] \wedge \omega \in [\omega_{\min}, \omega_{\max}]\}$$

式中，$v_{\min}$和$v_{\max}$分别为移动机器人的最大线速度和最小线速度，$\omega_{\min}$和$\omega_{\max}$分别为移动机器人的最大角速度和最小角速度。

b) 电机动力学约束

移动机器人电机的输出力矩是有限的，线速度和角速度的最大加速度和最大减速都受到限制。因此，在一个模拟周期内受到最大加减速度限制的速度集合V_d应满足：

$$V_d = \{(v, \omega) \mid v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \wedge \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t]\}$$

式中，v_c和ω_c分别为移动机器人当前实际的线速度和角速度，$\dot{v}_a$和$\dot{v}_b$分别为移动机器人线速度的最大加速度和最大减速度，$\dot{\omega}_a$和$\dot{\omega}_b$分别为移动机器人角速度的最大加速度和最大减速度。

c) 安全性约束

为了保证动态窗口法避障的安全性，采样的速度要能在移动机器人碰撞到障碍物之前停止运动，所以采样速度空间还应受到制动距离的约束。因此，在一个模拟周期内采样速度矢量在最大减速度制动下，能够让移动机器人与障碍物碰撞前停下来的速度集合 V_a应满足

$$V_a = \left\{(v, \omega) \mid v \leq \sqrt{2 \cdot \mathrm{dist}\,(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot \mathrm{dist}\,(v, \omega) \cdot \dot{\omega}_b}\right\}$$

式中，$dist(v, \omega)$为速度矢量(v, ω)对应轨迹上离障碍物的最近距离。

综合上述三个速度约束条件，最终可选取的速度即为上述三个集合的交集，令 V_r表示可容许速度集合，则V_r应满足：

$$V_r = V_s \cap V_d \wedge V_a$$

（2）计算出机器人在当前运动周期下所有采样速度组合(v, ω)下的移动轨迹。移动机器人在世界坐标系下的运动学位姿用可用向量$\mathrm{q} = [\mathrm{x(t)}, \mathrm{y(t)}, \theta(\mathrm{t})]$来表示，其中 $\mathrm{x(t)}$和$\mathrm{y(t)}$分别表示移动机器人在世界坐标系下 t 时刻的坐标，$\theta(\mathrm{t})]$表示移动机器人在 t 时刻的航向角。如图 3-3 所示，设 $\mathrm{v(t)}$表示移动机器人在 t 时刻的线速度，$\omega(\mathrm{t})$为的角速度。由于移动机器人在一个时间采样周期内移动机器人位移较小，所以在相邻两个时间采样点内将移动机器人的轨迹近似为匀速直线运动，那么移动机器人相邻时刻位置和航向角的增量可以表示为：

$$\begin{cases} \Delta x = v_t \Delta t \cos(\theta_t) \\ \Delta y = v_t \Delta t \sin(\theta_t) \\ \Delta \theta = \omega \Delta t \end{cases}$$

此时，$\mathrm{t}+1$ 时刻的移动机器人位姿可以表示为：

$$\begin{cases} x(t+1) = x(t) + v_t \Delta t \cos(\theta_t) \\ y(t+1) = y(t) + v_t \Delta t \sin(\theta_t) \\ \theta(t+1) = \theta(t) + \omega \Delta t \end{cases}$$

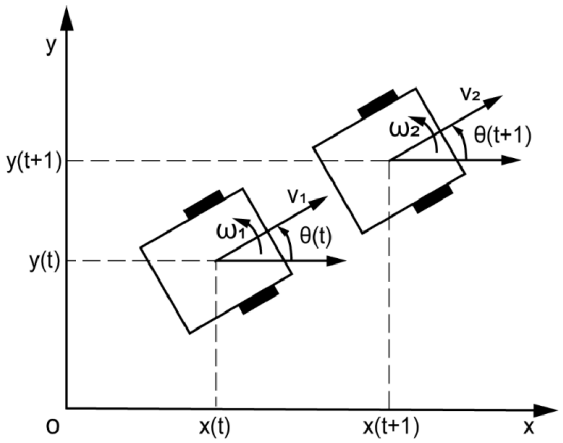


图 1.5.4.12 移动机器人运动学模型

（3）以当前障碍物分布情况、全局目标路径点位置和速度大小对每条候选移动轨迹进行评价。轨迹评价函数由目标方位角评价函数，障碍物间隙评价函数和速度评价函数三个加权项组成，其定义式如下：

$$G(v, \omega) = \varepsilon(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega), + \gamma \cdot vel(v, \omega))$$

式中，α、β 和 γ 分别为目标方位角评价函数、障碍物间隙评价函数和速度评价函数的权重，ε 为对轨迹评价函数的三个部分进行归一化处理。

下面分别介绍轨迹评价函数中的三个分量：

a) 目标方位角评价函数$heading(v, \omega)$

目标方位角评价函数$heading(v,\omega)$用来衡量移动机器人的模拟轨迹末端点朝向目标点的程度，如式（3-1 2）所示。如图 3-3 所示，在一个模拟周期内移动机器人所产生的模拟轨迹末端点朝向和目标点连线的夹角为θ。θ值越小，说明移动机器人更倾向于朝着目标点移动，此时的目标方位角评价函数就越大

$$heading(v,\omega) = 180^{o} - \theta$$

式中，θ为机器人模拟轨迹末端点朝向和目标点联系的夹角。

b) 障碍物间隙评价函数$dist(v,\omega)$

障碍物间隙函数$dist(v,\omega)$表示移动机器人模拟轨迹上的点到最近障碍物的距离，用来降低移动机器人与障碍物发生碰撞的概率。障碍物间隙函数值越大，表明在该模拟轨迹上移动机器人离障碍物越远，此时移动机器人导航的安全性越高。式（3-13）中d_{max}的作用是避免某一模拟轨迹上没有障碍物，导致障碍物间隙函数过大，在总的轨迹评价函数中占比过大。

$$dist(v,\omega) = \begin{cases} d & ,d < d_{max} \\ d_{max} & ,d \geq d_{max} \end{cases}$$

式中，d为移动机器人模拟轨迹上的点到最近障碍物的距离，d_{max}为移动机器人到障碍物距离的阈值。

c) 速度评价函数$vel(v,\omega)$

速度评价函数$vel(v,\omega)$用于评估机器人在向目标点行进过程中的前进速度，如式（3-14）所示。速度评价函数值越大，表明在该模拟轨迹上移动机器人能更快靠近目标点，但过大的速度也会导致移动机器人在离目标点很近时错过目标点。

$$\mathrm{vel}(v,\omega) = | v_g |$$

式中，v为移动机器人模拟轨迹对应的线速度。

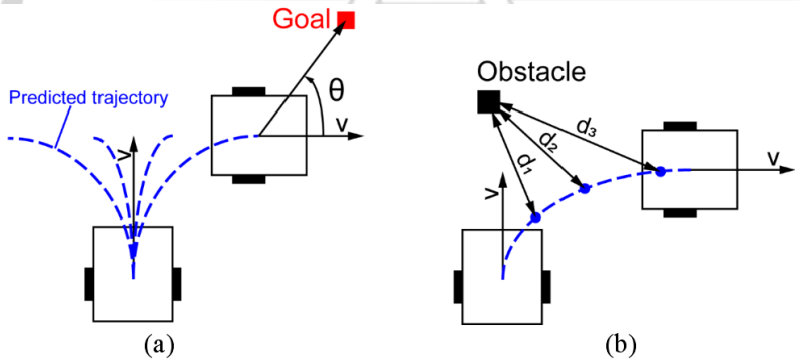


图 1.5.4.13 目标方位角评价函数和障碍物间隙评价函数示意图：

（a）机器人朝向与目标点的角度θ；（b）移动机器人模拟轨迹上的点到最近障碍物的距离d。

得到上述三个部分的评价函数结果后要将每个部分做归一化处理再相加，其目的是为了使轨迹更平滑，避免某一部分占比过大。归一化的过程就是令每一项除以每一项的总和，使轨迹评价函数的三个组成部分都被归一化到区间[0,1]，归一化计算公式为

$$normal\_heading(i) = \frac{heading(i)}{\sum_{i=1}^{n} heading(i)}$$

$$normal\_dist(i) = \frac{dist(i)}{\sum_{i=1}^{n} dist(i)}$$

$$normal\_vel(i) = \frac{vel(i)}{\sum_{i=1}^{n} vel(i)}$$

其中，n为在一个模拟周期内的所有采样轨迹，i为当前待评价的模拟轨迹。

（4）选取评价得分最高的那条移动轨迹，将其对应的线速度和角速度经过坐标系变换后（从世界坐标系变换到底盘坐标系）发给下位机。

3.算法性能

3.1 全局路径规划

算法优点：

经过二次优化后的全局路径规划算法已经通过物理仿真测试和真实场景测试，得到的全局路径具有安全高效的特点。

与传统 A 算法相比，改进算法所获取的路径点个数是 A 算法获取路径点个数的 1/20~1/30，相邻路径点的转角幅度更小，并且在保障路径效率的同时降低了移动机器人与环境障碍物碰撞的概率。

需改善点：

第一次路径优化的时间较长。第一次路径优化的时间消耗主要用于判断两个路径点连线间是否存在障碍物，该检测算法由于将连线的采样点个数和地图分辨率挂钩，在提高地图分辨率的时候该时间消耗也会大幅提升，因此可以考虑在该环节提升检测效率以降低路径优化的时间消耗。

存在移动障碍物时的路径重规划。最开始采用的方案是定时重规划，然而在绝大多数情况下（没有临时障碍物阻挡移动）是不需要重规划全局路径的，这种定时重规划就成为了一种不必要的算力开销。后来我们引入激光雷达的局部点云信息来对重规划做判断，如果检测到当前全局路径上存在障碍物点云再进行重规划，该方案对静态临时障碍物的规避效果较好。针对移动方向存在动态障碍物的情况下，如果对方运动轨迹较为复杂，则可能出现路径方向

频繁变化的情况，路径方向的频繁变更可能会降低机器人的移动效率，因此需要对重规划算法继续加以改进。

提高栅格地图分辨率。为了提升障碍物碰撞检测精度，需要尽可能高的地图分辨率。目前激光雷达的局部点云反馈频率为$10hz$，因此规划算法的执行周期至少要小于$100ms$，目前如果要满足全场路径规划则只能将地图分辨率提升到$0.1m$的量级，为了应用更高的地图分辨率，需要对规划算法的各个函数进行针对性优化。

3.2 局部轨迹规划

算法优点：

改进 DWA 算法将障碍物距离、全局路径点方向、动力学约束和运动学约束等因素均以评价函数的方式引入轨迹设计中，可以针对所应用场景和机器人特点定制化速度评价函数，从而设计出便于机器人执行的移动轨迹。

可以将视觉系统识别出的多目标移动轨迹加入哨兵机器人局部轨迹设计中，从而使哨兵机器人的移动具备一定的前瞻性，从而提高针对动态障碍物的躲避能力。

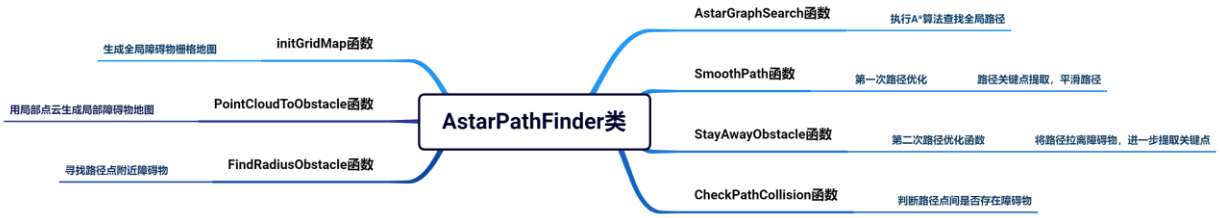
局部轨迹规划的执行时间为$2\sim3ms$，可以实时评价 200 条由不同线速度和角速度搭配生成的移动轨迹。

需改善点：

轨迹规划的输出结果是机器人的实时移动线速度和角速度，哨兵机器人采用的是舵轮底盘，而舵轮底盘对线速度和自转角速度的控制效果不佳，从而导致机器人机动能力上限不高。因此目前针对轨迹规划有三种优化方案：（1）将局部轨迹规划的规划对象由线速度和自转角速度更改为线速度和航向角。（2）对现有舵轮底盘进行优化设计以提高自转角速度控制能力；（3）将哨兵机器人的底盘形式更改为麦轮底盘。

4.算法库介绍

4.1 全局路径规划



InitGridMap 函数：使用三维点云地图降采样生成全局障碍物栅格地图。

PointCloudToObstacle 函数：使用激光雷达实时点云更新局部栅格地图。

AstarGraphSearch 函数：执行 A*算法查找全局路径。

SmoothPath 函数：第一次路径优化，路径关键点提取，平滑路径。

StayAwayObstacle 函数：第二次路径优化，将路径拉离障碍物，进一步提取关键点。

CheckPathCollision 函数：判断路径点间是否存在障碍物。

FindRadiusObstacle 函数：寻找路径点附近障碍物。

4.2 局部路径规划

CalculateTrajectoryEndStates 函数：计算所有速度组合的移动轨迹。

CalculateTrajectoryScores 函数：计算所有候选移动轨迹的得分。

getTrajectoryHeadingScore 函数：计算轨迹朝向得分。

getTrajectoryDistanceScore 函数：计算障碍物间隙得分。

getTrajectoryVelocityScore 函数：计算速度得分。

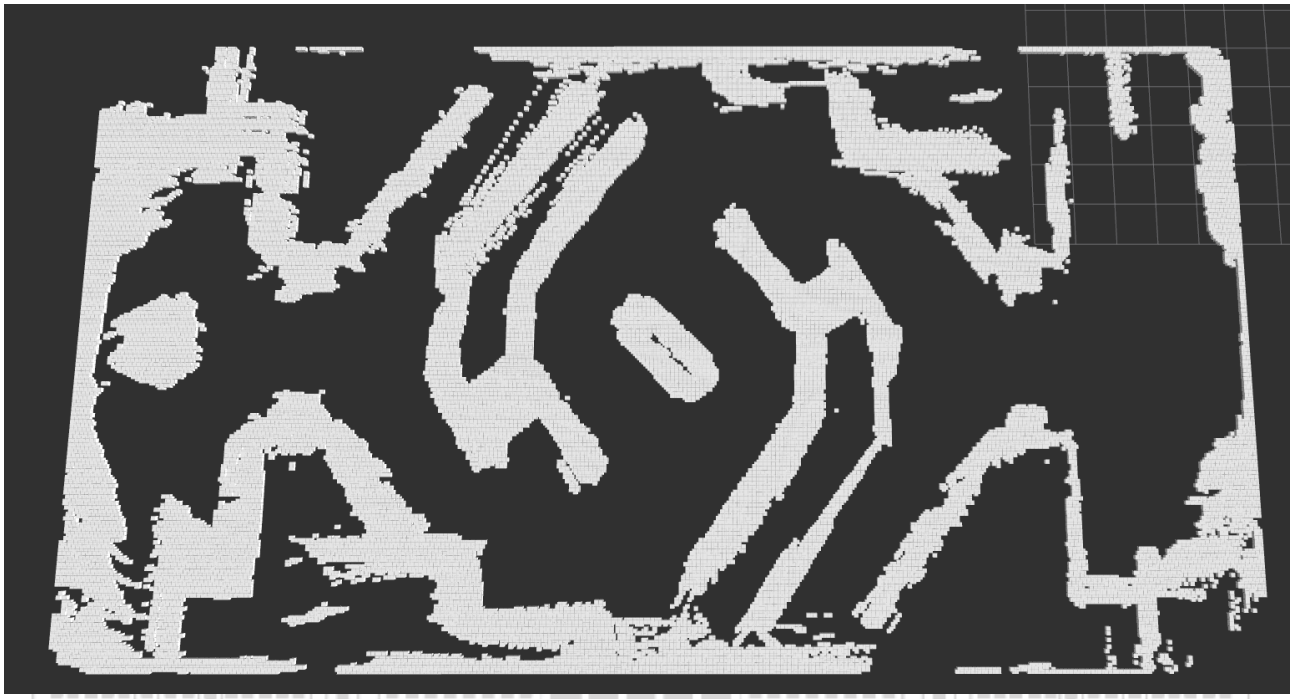
getOptimalTrajectory 函数：选取得分最高的移动轨迹。

getTrajectoryPoints 函数：对所有参考轨迹进行采样。

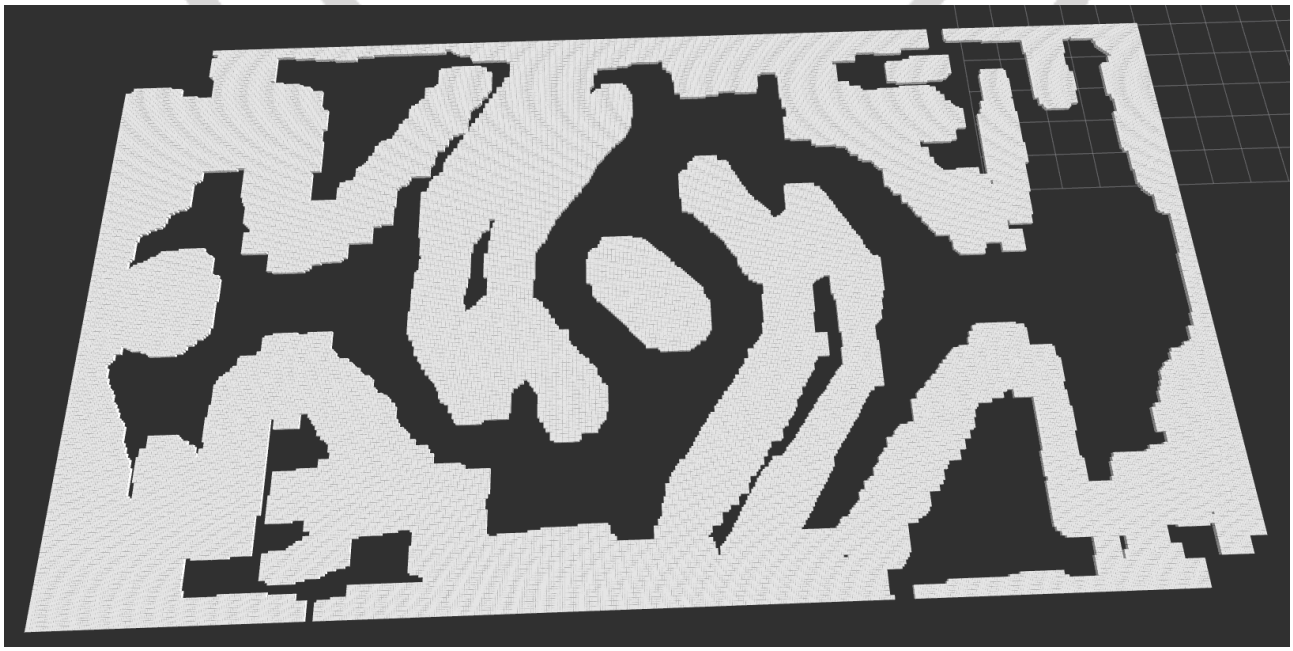
5.算法结果展示

5.1 全局路径规划

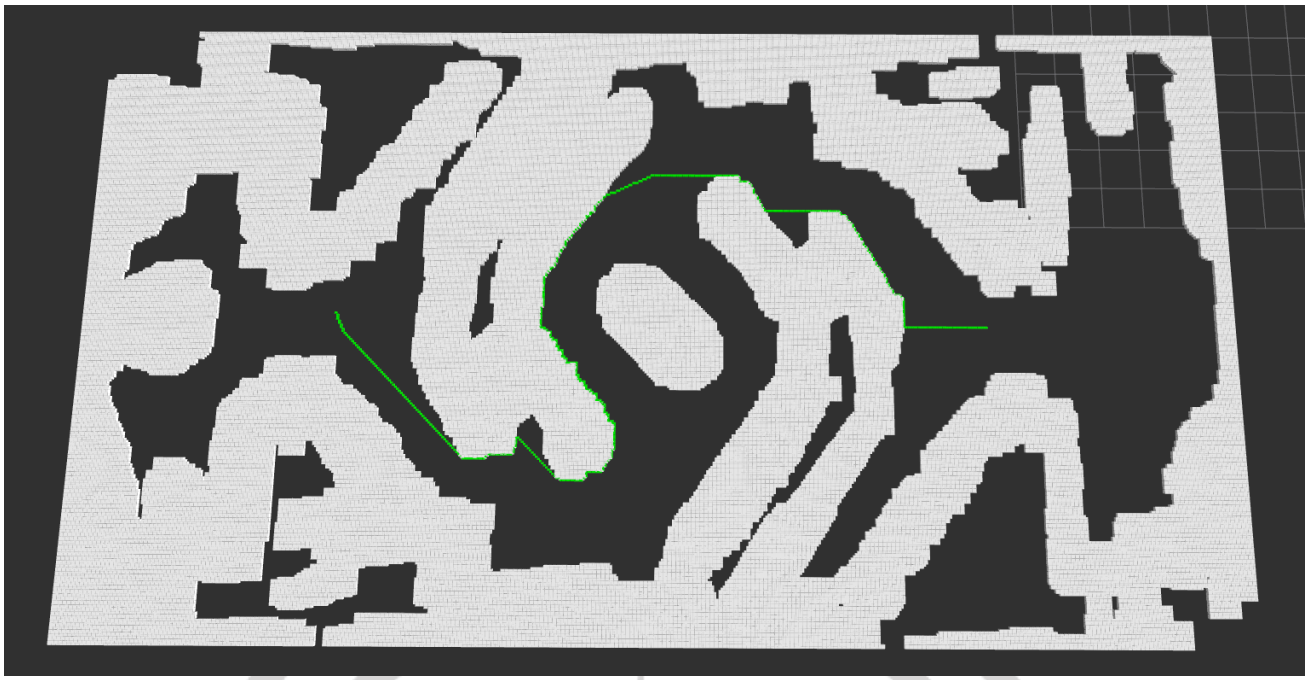
（1）对激光雷达获取的三维全局地图进行降采样获取二维栅格地图



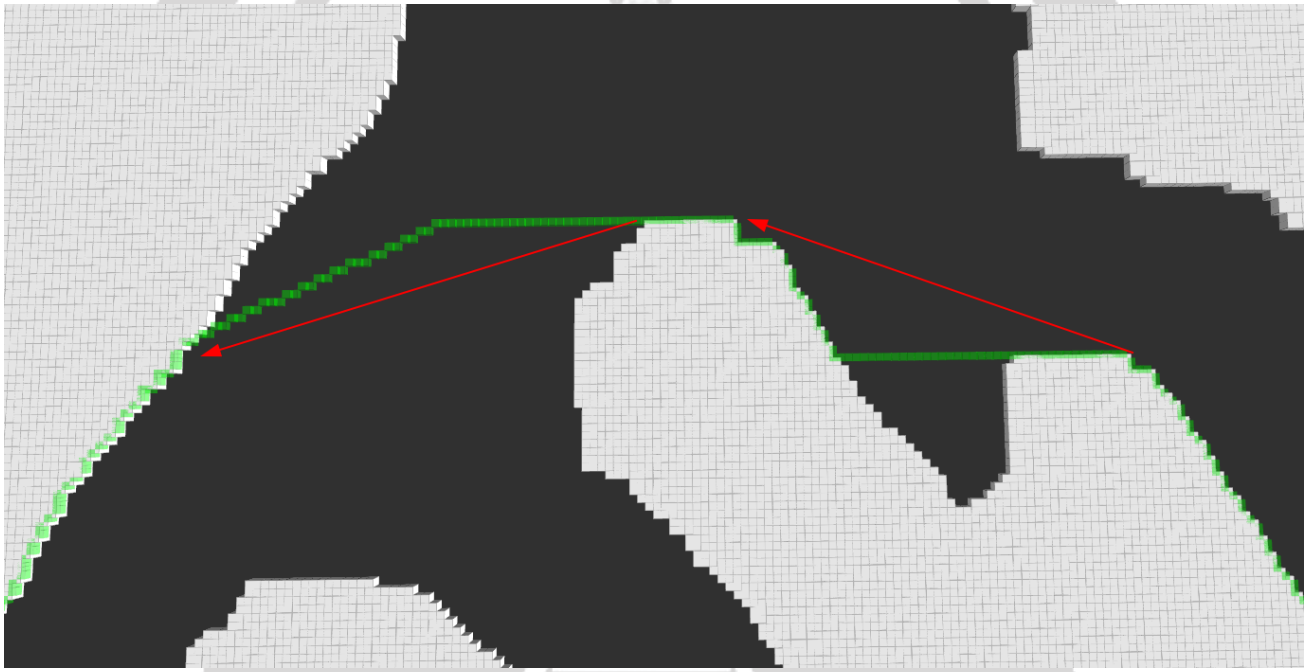
（2）对栅格地图进行膨胀处理



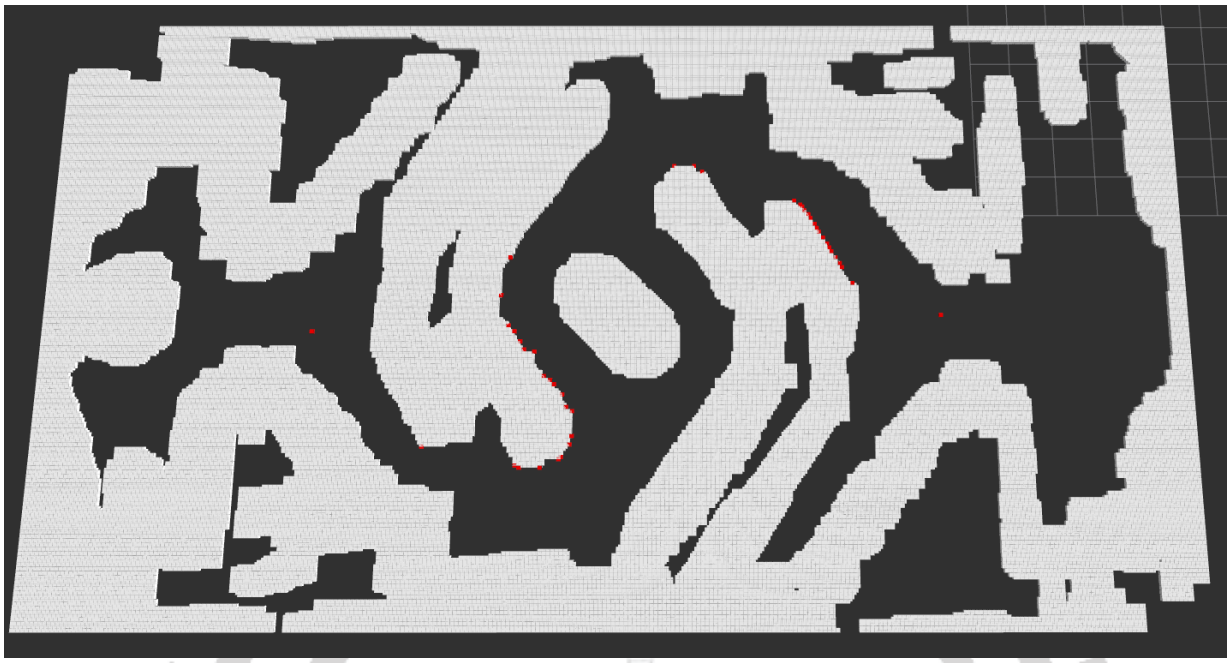
（3）执行 A*算法查找全局路径。



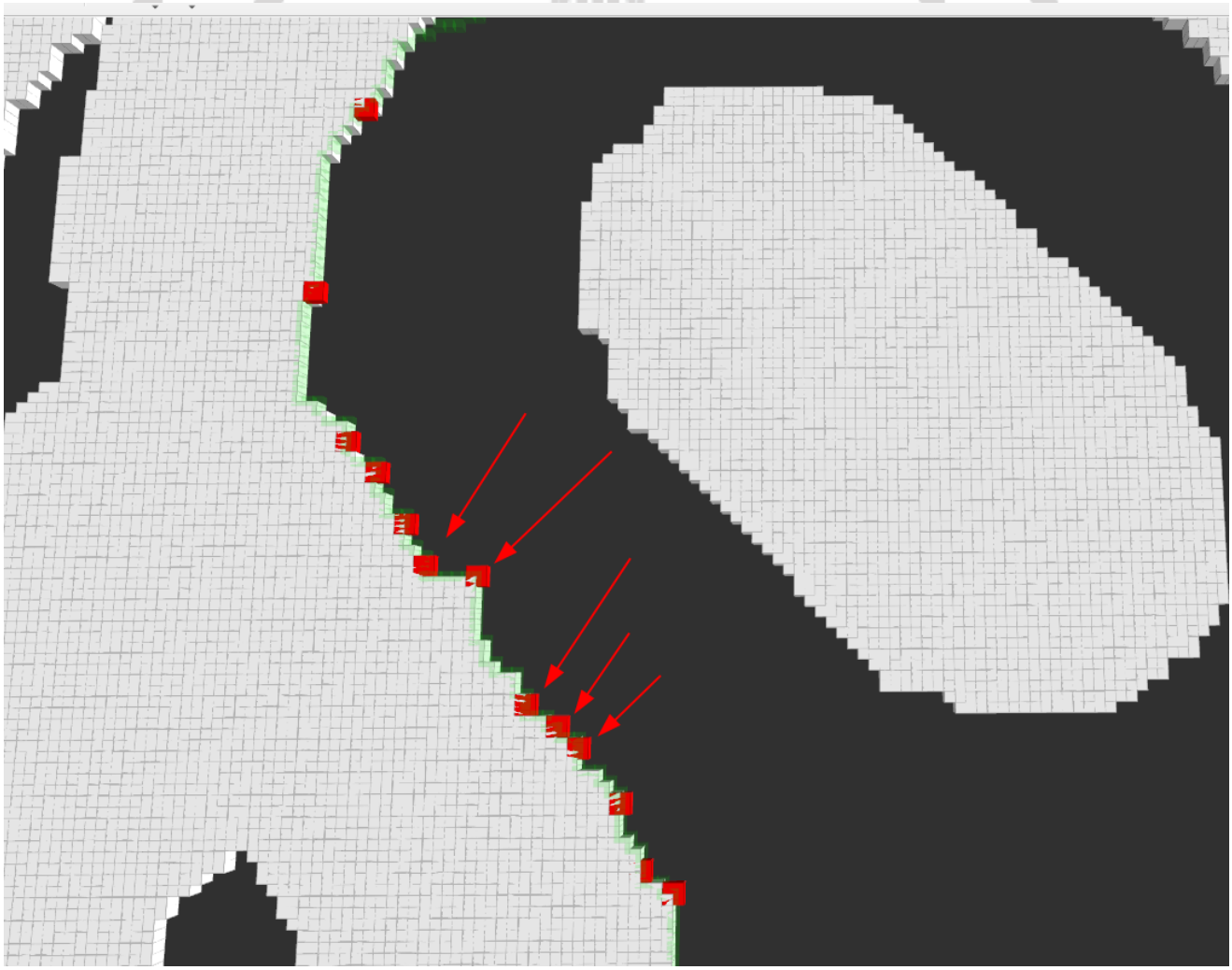
（4）存在冗余路段



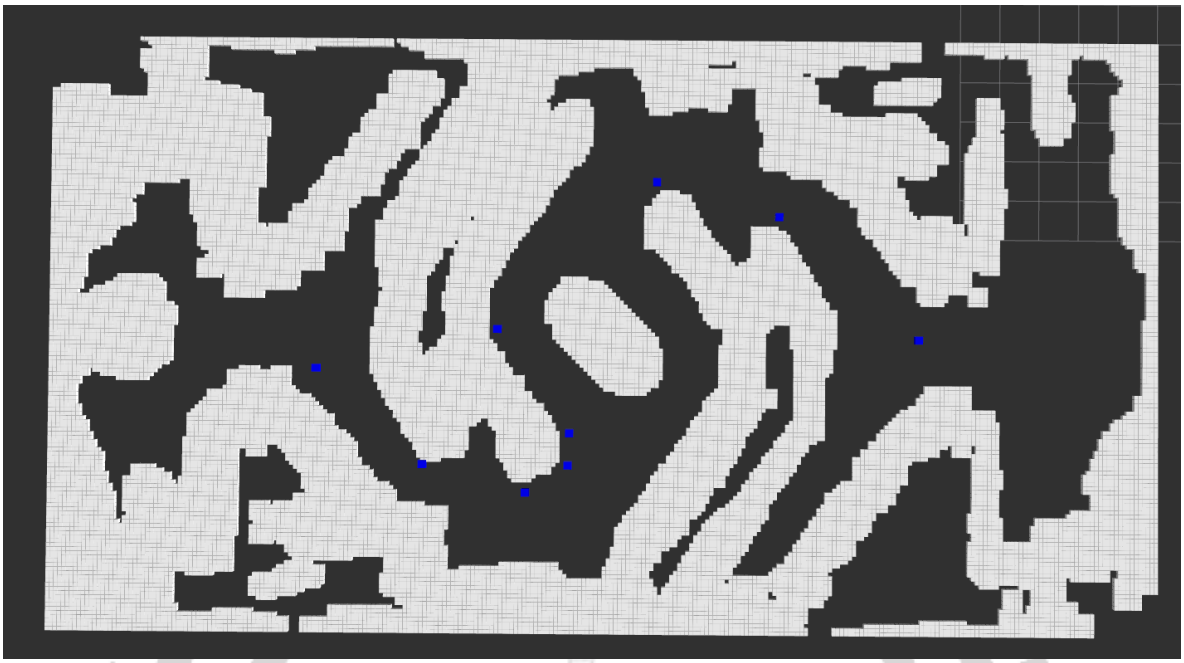
（5）第一次路径优化提取关键路径点消除冗余路段



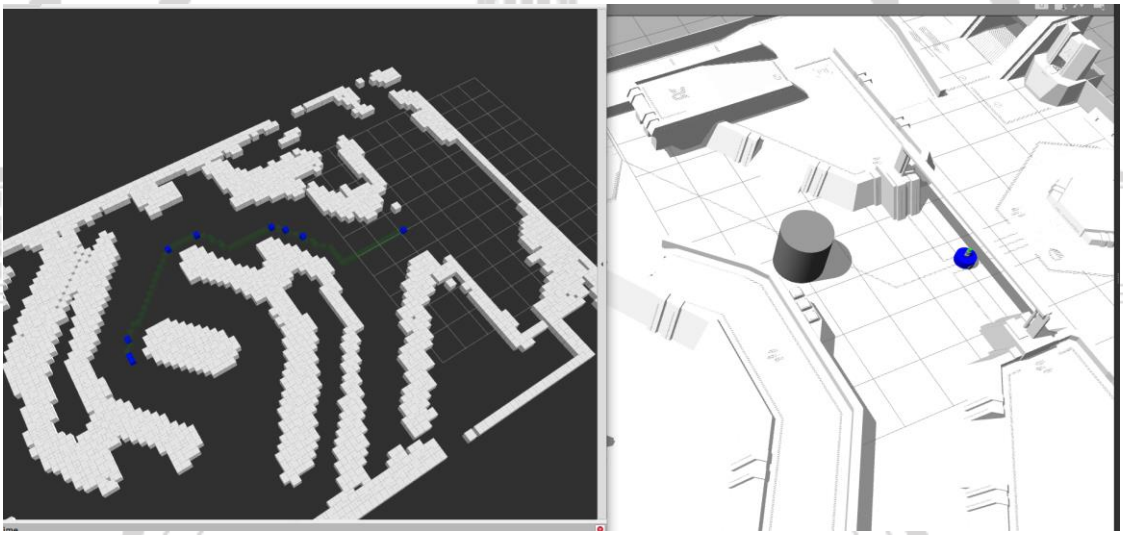
（6）障碍物附近路段不够平滑

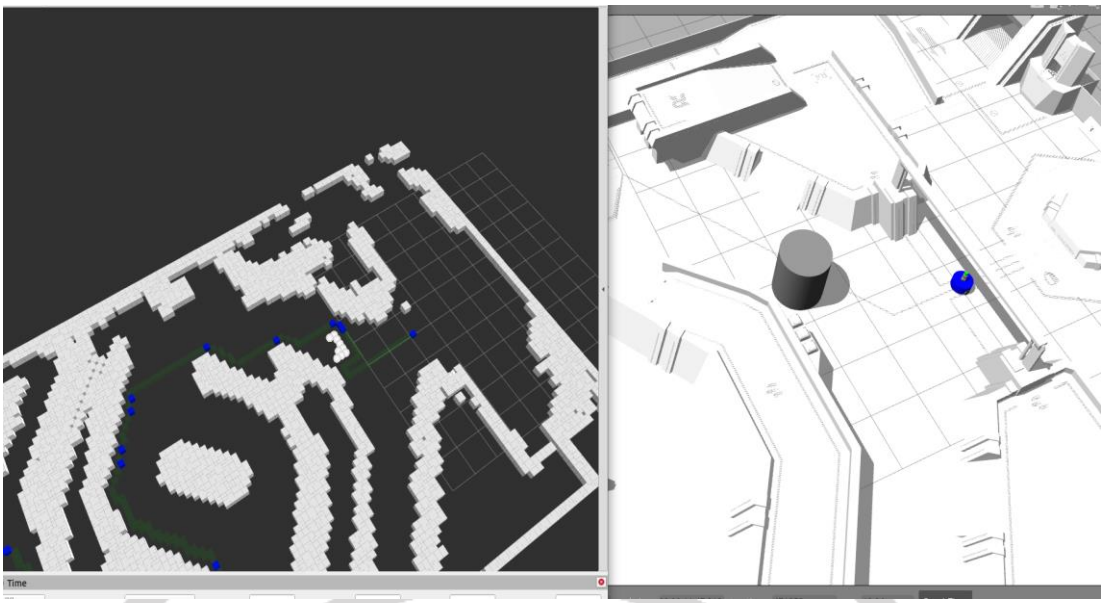


（7）第二次路径优化，将路径拉离障碍物，进一步提取关键点



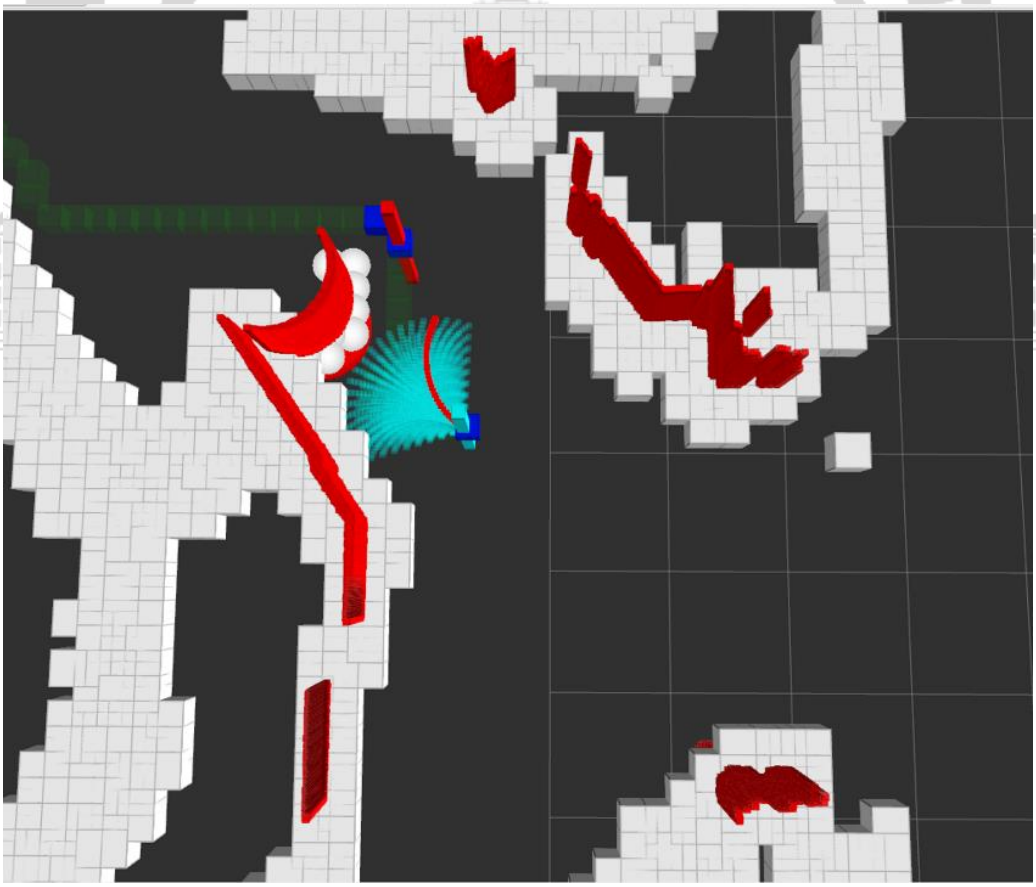
（8）检测到临时障碍物时执行重规划。





5.2 局部轨迹规划

（1）计算当前速度空间中所有速度组合的移动轨迹，选择最优移动轨迹



1.5.4.3 决策算法

决策器的主要决策输出为：为路径规划模块提供目标位置用于控制机器人的运动，为辅瞄模块提供攻击的目标。

场地稀疏顶点图

为了降低计算的复杂度，加快决策速度，对场上位置进行抽样，决策器在选定目标位置时只需在这些具有典型性位置中进行选择即可。抽样点主要选择原则为：

（1）具有加成效果的位置。

（2）高地等可以隐藏车体的区域后面

（3）具有开阔视野范围的火力输出点。

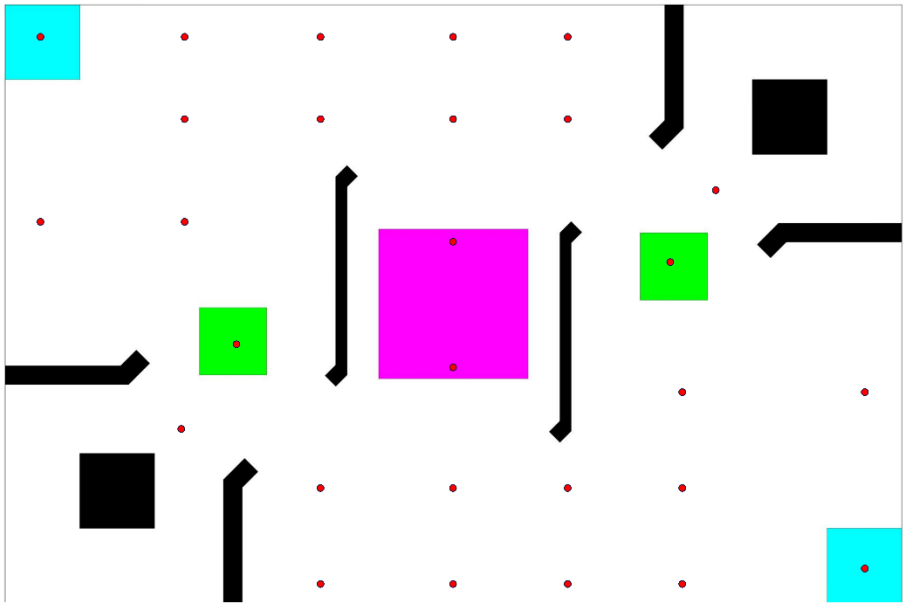


图 1.5.4.14　场地稀疏顶点图(省赛地图版)

决策模式

哨兵决策根据不同的任务需求，划分出不同的决策模式，目前有以下三种模式：

（1）特定任务模式。在该模式下，哨兵会根据提前制定好的任务轨迹，完成阶段性的任务需求。

（2）巡逻模式。在该模式下，哨兵依据位置决策进行游走，完成不同的击打任务。

（3）Moba 模式。在该模式下，哨兵和操作手进行交互，完成操作手给定的击打任务。

位置决策

为了选择当前条件下最佳的目标位置点，设计了位置权重函数用于判断当前地图上各个顶点的权重，进而根据权重选择目标位置。

1 位置权重函数

$$\mathrm{F}(N)=\prod_{i}\eta(d_i)\,\sigma_{\text{self-i}}\sigma_{\text{i-self}}+\phi_{\text{support}}$$

其中，$\prod_i \eta(d_i)\,\sigma_{\text{self-i}}\sigma_{\text{i-self}}$为打击火力权重，$\phi_{\text{support}}$为己方援助权重。

2 位置权重函数各部分介绍

(1) 打击命中率分布函数

设计原则：

该函数根据实测的命中率进行近似设计，其中当距离小于$1m$时，由于视野问题以及 pitch 限位问题，打击命中率较低；当距离在$1m$~$3m$距离范围内，命中率较高；当距离超过$3m$时，打击命中率会随着距离增加而降低，在这里做了线性近似的处理。

函数形式：

$$\eta(d)=\begin{cases}0, & d<1m\\ 1, & 1m<d<3m\\ 1.6-0.2d, & d>3m\end{cases}$$

(2) 有效打击区域判定函数

设计原则：

该函数主要考虑哨兵在击打敌方的同时需要保证自身的生存能力，因此将打击区域划为：敌方 i 在有效打击范围内、敌方 i 不在有效打击范围内、己方在敌方有效打击范围内以及己方不在敌方有效范围打击内；

函数形式：

$$\sigma_{\text{self-i}}=\begin{cases}2, & \text{敌方 i 在有效打击范围内}\\ 0.5, & \text{敌方 i 不在有效打击范围内}\end{cases}$$

$$\sigma_{\text{i-self}}=\begin{cases}2, & \text{己方不在敌方 i 有效打击范围内}\\ 0.5, & \text{己方在敌方 i 有效打击范围内}\end{cases}$$

(3) 己方援助函数

设计原则：

该函数主要考虑联合队友进行行动，在队友附近的顶点优先级比较高，远离队友的顶点优先级应随着距离逐步降低。

函数形式：

$$dis = \sum_{i} \text{Euclidean Distance}(i, \text{哨兵})$$

$$\phi_{\text{support}} = C_1 \cdot \text{sigmoid}(\text{dis})$$

(4) 节点屏蔽策略

由于决策分为不同的模式，每个模式下所执行的任务不同，因此在不同模式下位置决策所需要的位置节点是不同的，需要屏蔽掉不需要考虑的节点。如在巡逻模式中，在己方前哨战还未被击败的情况下，位置决策会考虑场上所有的位置顶点；当己方前哨战被击败时，哨兵将回退到巡逻区域进行巡逻，此时位置决策会屏蔽所有非巡逻区的顶点。

3 规则决策器的架构

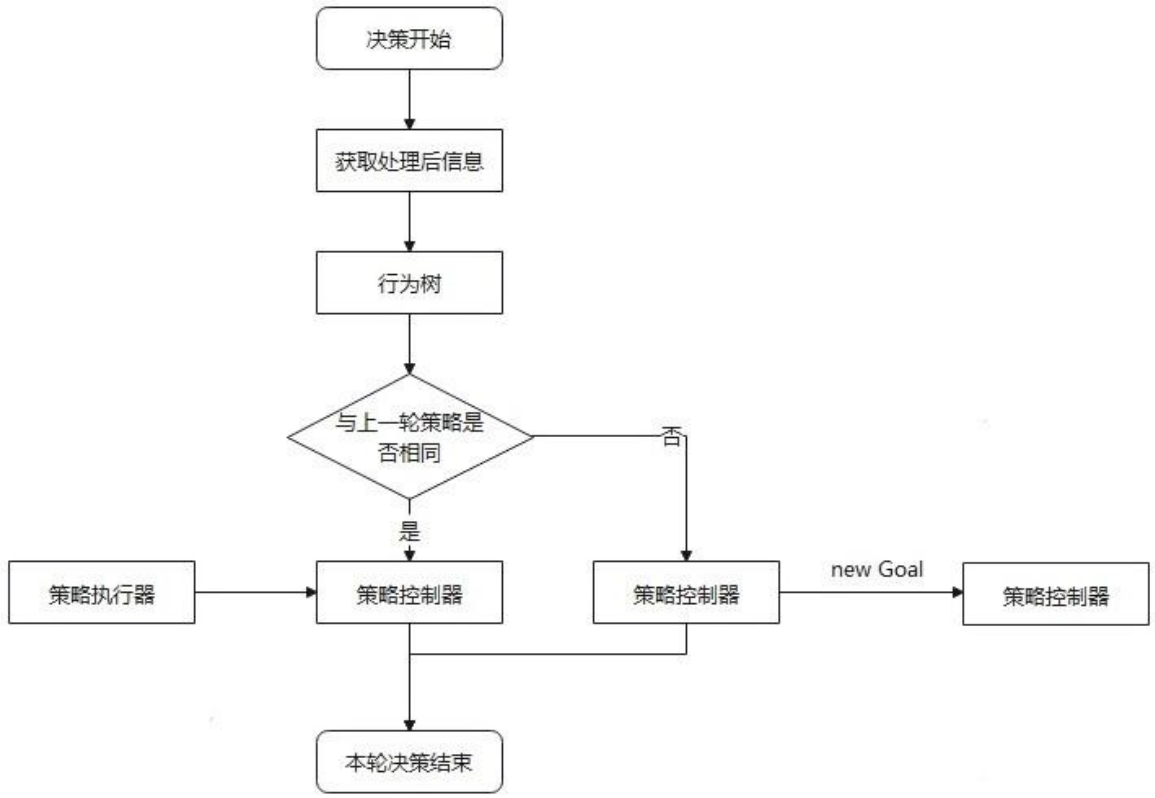


图 1.5.4.15　规则决策器架构

决策模拟器

为了保证决策逻辑的正确性，我们构建了一个模拟器用来测试决策对底盘控制的执行逻辑是否符合我们的预期。

与实际的测试环境相比，仿真环境具有速度快，成本低等优点，通过在仿真环境中搭建

简化模型，例如将比赛场地简化为二维平面，将机器人也简化为二维，简化去除对于决策影响不大的因素，例如车体的结构等，从而获得一个能够快速运行仿真的环境，提高训练的速度与效率。

同时，为了获取更多样的数据，模拟器还具有联机的功能，可以让多位操作手进行操作，确保决策逻辑能在更加逼真的场景内多次测试。下图是决策模拟器的软件架构。

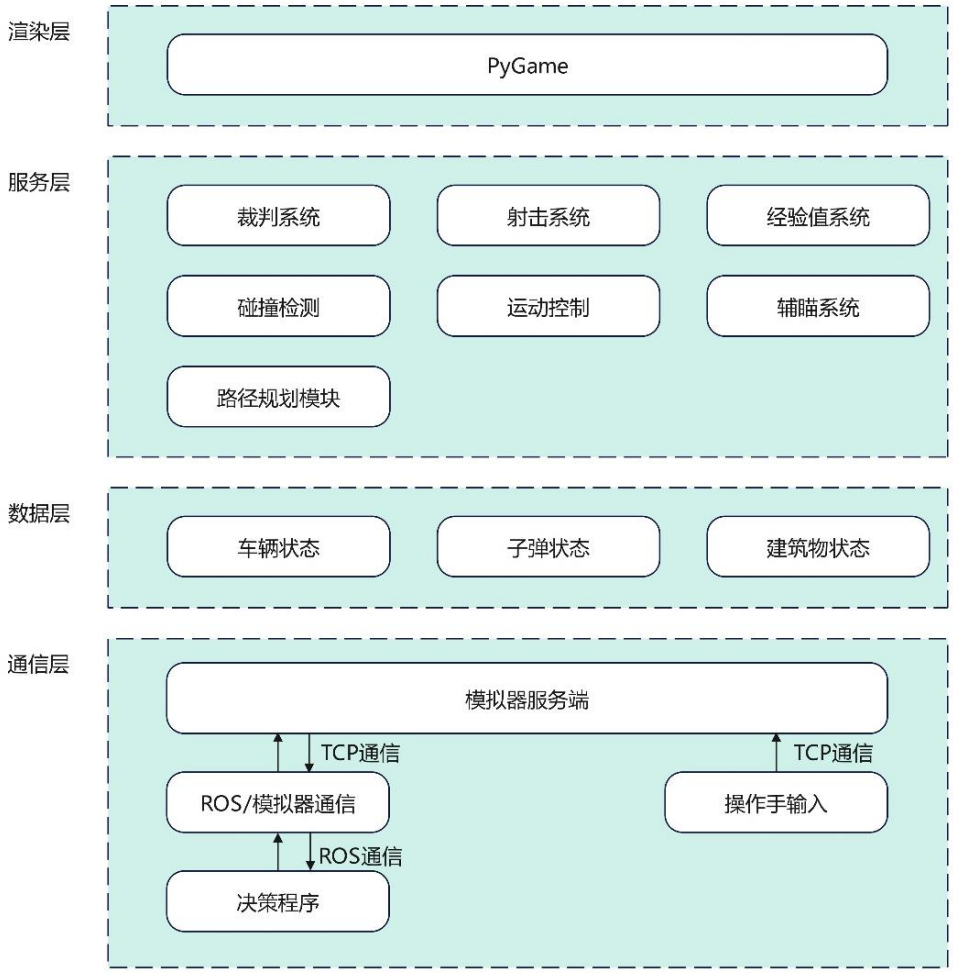


图 1.5.4.16　决策模拟器软件架构

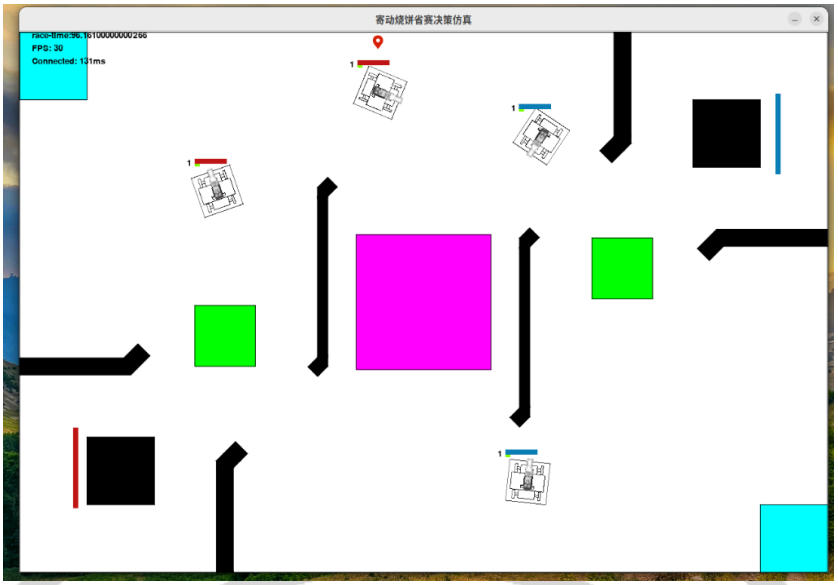


图 1.5.4.17　决策模拟器截图(省赛地图版)

1.5.4.4 反陀螺识别

光流法

光流（optical flow）是空间运动物体在观察成像平面上的像素运动的瞬时速度。光流法是利用图像序列中像素在时间域上的变化以及相邻帧之间的相关性来找到上一帧跟当前帧之间存在的对应关系，从而计算出相邻帧之间物体的运动信息的一种方法。如图 1.5.4.28 所示，静止的观众光流矢量几乎为 0，运动员光流矢量指向其运动方向。

图 1.5.4.28　光流图式

对于比赛的机器人而言，其陀螺运动与平移运动是有很大的区别的，我们提出了一种通过光流来区分机器人运动的方法。我们通过相邻帧的装甲板光流特征，投入 MobileNet 卷积神经网络进行训练分类，以区分机器人是否正在进行陀螺运动，为之后的运动解算，辅瞄预测做准备。装甲板光流特征如图 1.5.4.29 所示，神经网络测试结果如图 1.5.4.30 所示。

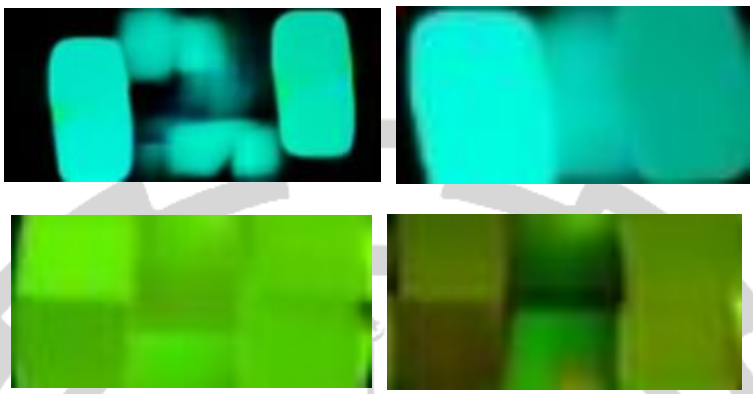


图 1.5.4.29 装甲板光流

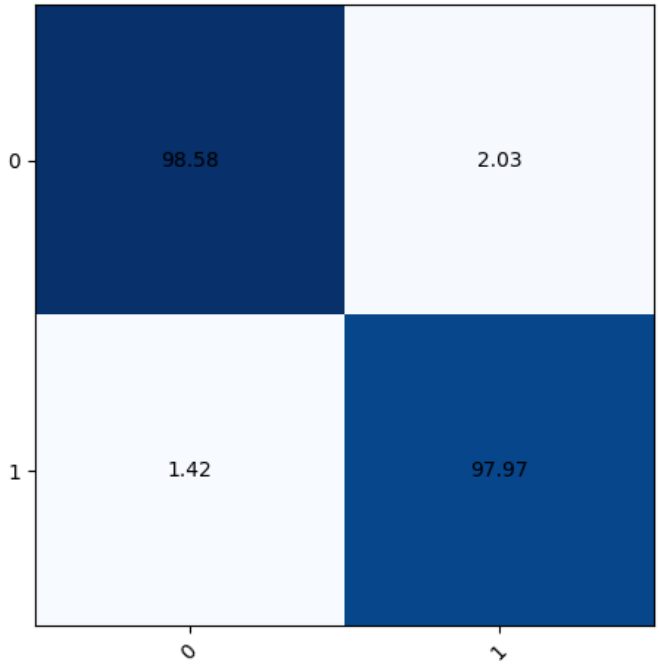


图 1.5.4.30　MobileNet 分类结果

经过实车测试，分类的精度较高，基本实现了小陀螺运动识别的功能。在实际使用中表现较为稳定，在稠密光流中普通运动和小陀螺运动在视觉直观上区别较大，用这种方法进行分类直接有效。

关键点序列神经网络

RoboMaster 对抗赛中小陀螺是最常见的运动状态之一，小陀螺运动会对视觉辅瞄系统造

成显著的困难，因此对敌方目标的小陀螺运动状态进行有效识别是视觉上一个重要的问题。机器人陀螺运动显然与时间强相关，因此我们采用 LSTM 序列网络对装甲板运动状态进行拟合分类。LSTM 网络单元如图 1.5.4.31 所示。

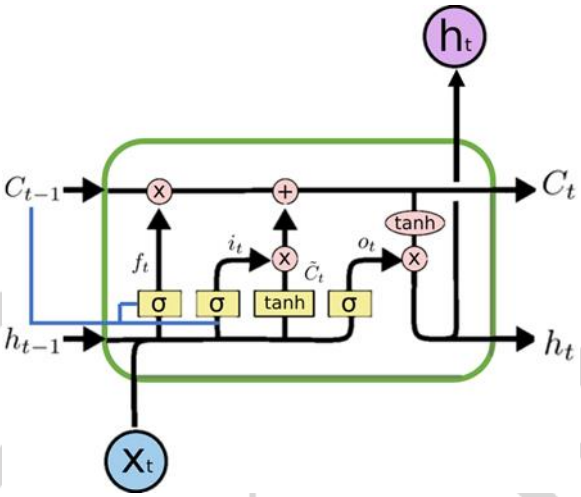
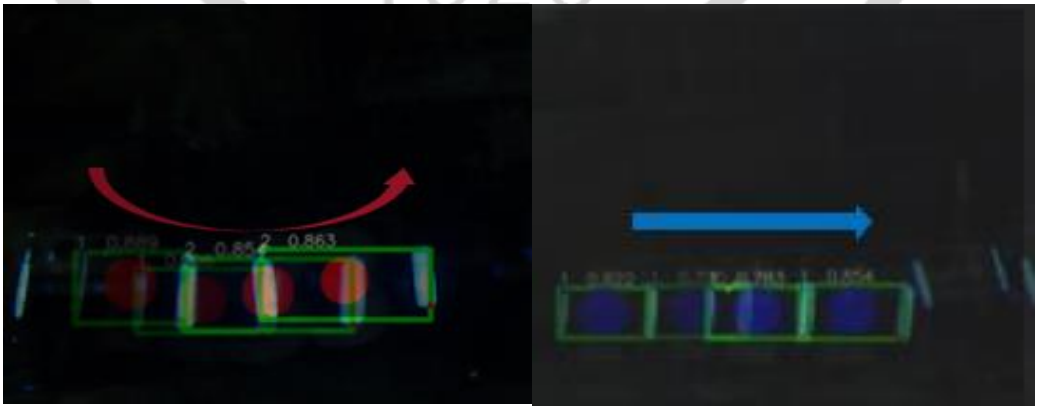


图 1.5.4.31　LSTM 网络单元

把小陀螺运动抽象成序列，就可以利用序列模型对目标是否正在小陀螺的行为进行分类。模型的数据输入可以选择 2D 图像数据或者 3D 解算的数据，3D 的数据需要经过解算系统会带有系统噪声，2D 的噪声较小但是有一定信息损失。我们认为图像上装甲板的四个角点有几何约束关系并且可以映射装甲板的法向量，所以将四个角点全部输入网络，经过实验验证 2D 图像关键点的数据输入足够进行小陀螺分类，在测试集和实车验证上取得了较好的效果。

实验测试分类结果准确率可达 99%以上。关键点序列分类结果如图 1.5.4.32 所示。



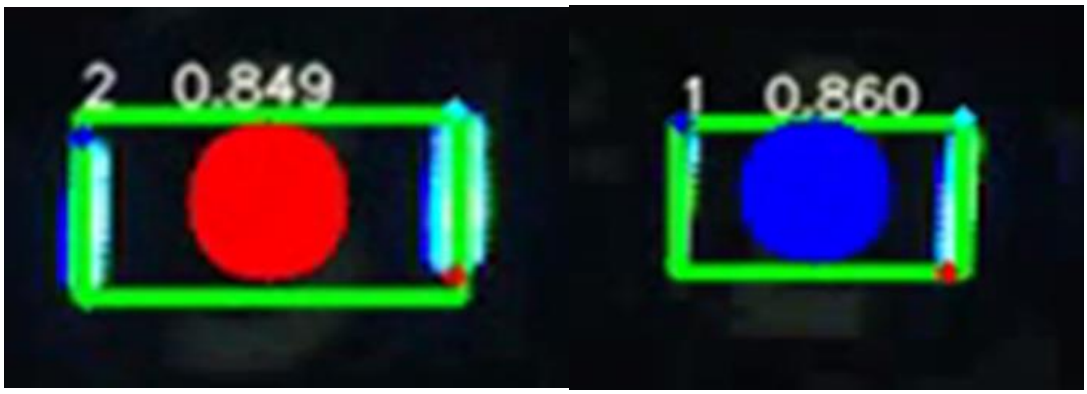


图 1.5.4.32 检测结果，红色代表陀螺运动，蓝色代表非陀螺运动

1.5.5 其它

基于 Notion 的管理方法

Notion 是一个用于工作管理或者日常生活行动管理的网站，拥有网页、PC 端软件、手机端软件（Android、IOS）等不同运行环境，可以实现在任意设备上的工作管理，并且所有运行环境都是实时同步的。在此基础上，Notion 还提供了团队管理，方便项目团队进行在线协作办公，提高交互效率，促进工作的有效推进。下面是哈工大竞技机器人队 2023 哨兵组使用 Notion 进行研发管理的具体方案。

在【信息同步】界面，会显示当前日历。各模块研发负责自行填写当前任务，预估其工时与完成日期规划。技术指导会定期查看进度，并在截止日进行验收相关动作（图 1.5.5.1 左）。

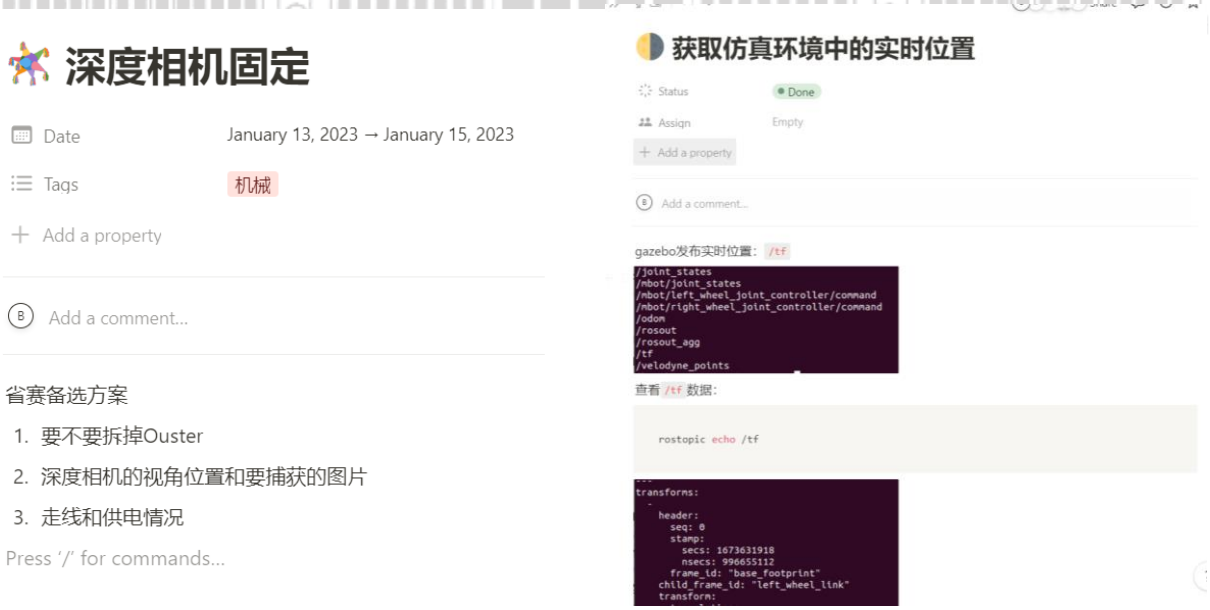


图 1.5.5.1 左：Notion 日历任务信息同步界面 深度相机固定任务安排 内容填写示例
右：Notion 任务管理界面 仿真环境位置获取情况 内容填写示例

在【任务管理】界面，会显示当前各模块的主进度。此部分由各技术栈指导填写，简要描述当前技术栈完成情况及遇到的问题。例如联调 bug、合代码的接口需求等（图 1.5.5.1 右）。

完成后将完成的部分证明（例如图片、代码截图等）上传。

在【方案论证】界面，会显示一些与方案制定相关的信息。例如老学长等其他指导提供的参考意见、部分备选方案的讨论区等。议题会以界面的形式呈现在 kanban 上，所有人可以进行更改和评论。（图 1.5.5.2）

在【技术文档】界面，会存放前期学习各技术栈时所写的技术文档及一些已实现的工作的记录。包括但不限于各类学习资料（教材、网课、模型、代码等）、各类技术文档（仿真教程、搭建框架、软件使用说明等）、各类测试记录（废弃方案测试、备选方案测试等）（图 1.5.5.3）

图 1.5.5.2 Notion 方案论证界面 云台方案讨论 内容填写示例

打开需要导出的模型，注意把所有不需要导出的部分全部压缩【注意不是隐藏】，同时方向配正。**注意把雷达删除掉，但要记住雷达中心安放的位置坐标。**gazebo中自带部分雷达的模型和功能，需要后续手动添加在导出的模型上。

坐标系需要提前自行定义，**按照gazebo中的世界定义，Z轴朝上，x轴朝前，右手系。在sw中借助构建轴和点的形式完成坐标系。**并及时合理命名。

物理量单位定义需要修改为MKS，因为gazebo里面长度尺度为米。

图 1.5.5.3 Notion 技术文档界面 仿真模型导出教程 内容填写示例

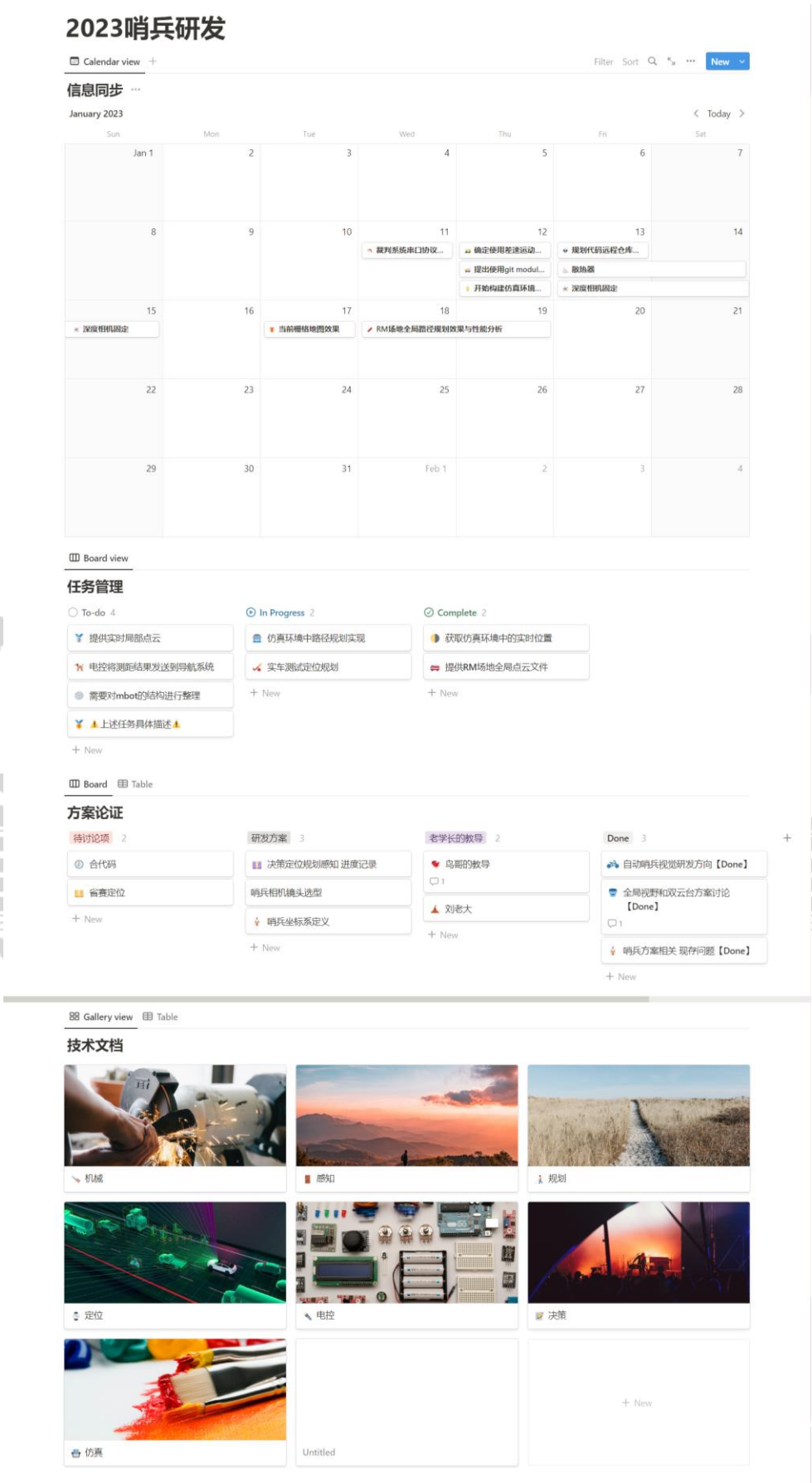


图 1.5.5.4 Notion 哨兵组页面总览

1.6 研发迭代过程

1.6.1 测试验收标准

云台调参验收标准：云台十分钟连续使用零漂肉眼不可见；Yaw/pitch 轴电机5°阶跃响应<130ms；

弹速验收标准：裁判系统端读数弹速波动<1.5m/s；连发 500 发无超射速情况；

机械弹道验收标准：使用参数整定后的云台击打静靶，7m小装甲板命中率> 90%；

辅瞄偏置验收标准：使用开启辅瞄的云台击打静靶，5m小装甲板命中率100%；

辅瞄调参验收标准：使用开启辅瞄的云台击打2m/s匀速运动动靶，5m 小装甲板命中率90%；

辅瞄陀螺验收标准：使用开启辅瞄的云台击打2rad/s匀速旋转动靶，5m 小装甲板命中率 70%。

1.6.2 版本迭代过程记录

版本号	功能或性能详细说明	完成时间
V1.0	一代车机械出车，交付电控布线调试	2023.1.7
V1.1	一代车电控初步调试完成，交付导航调试	2023.3.9
V1.2	一代车导航侧调试完成，交付决策调试	2023.4.11
V2.0	二代车机械出车，交付电控布线	*预计 2023.5.1*

1.6.3 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*	*解决人员*
1	调好的弹道只能保持 3 小时左右，或调试途中弹道突然散布变大	摩擦轮起皮严重	尝试更换了同硬度另一个厂家的摩擦轮。另外定时打磨摩擦轮。	V1.0、V1.1	机械组:：姚宇轩
2	全局规划在机器人处于障碍物内部时会规划失败	因为对障碍物进行膨胀	如果机器人中心所处的栅格在障碍物内部时进行判断，	V1.2	算法组：赵振廷

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	解决人员
			如果在一定范围内有可行点则将机器人的下一运动目标点设置为搜索到的最近可行点并进行重规划，否则进行其他运动		
3	全局规划频繁重规划会导致运动不连续	设置全局路径规划为定时重规划，导致运动过程中目标 yaw 值的不断变化。	将重规划条件设置为当前规划的最优路径间如果出现了新的障碍物或车辆的控制误差过大再进行重规划	V1.2	算法组：赵振廷
4.	数字识别同一种数字在大板和小板上表现的效果不同.	roi 截取算法依赖于装甲板尺寸，而大小板尺寸不同导致截取同一种数字之后数字在截图中的宽高比例也不同。	加入大小板的数据集，同时加入强数据增强提高网络鲁棒性，最终效果正常。	V1.1、V1.2	视觉组：韩镐轩

1.7 参考文献

[1] RoboMaster 2023 机甲大师超级对抗赛比赛规则手册 V1.2[EB/OL] .
https://terra-1-g.djicdn.com/b2a076471c6c4b72b574a977334d3e05/RM2023/RoboMaster%202023%20%E6%9C%BA%E7%94%B2%E5%A4%A7%E5%B8%88%E8%B6%85%E7%BA%A7%E5%AF%B9%E6%8A%97%E8%B5%9B%E6%AF%94%E8%B5%9B%E8%A7%84%E5%88%99%E6%89%8B%E5%86%8CV1.2%EF%BC%8820230331%EF%BC%89.pdf,2023-3-31
[2] RM2021-桂林电子科技大学双云台哨兵机械开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12303,2021-8-31.
[3] RMUC 2022 技术答辩丨浙江大学 Hello World 战队[EB/OL] .
https://www.bilibili.com/video/BV1tV4y137jn/?spm_id_from=333.788,2022-11-02.
[4] Controls Engineering in the FIRST Robotics Competition[EB/OL] .https://file.tavsys.net/control/controls-engineering-in-frc.pdf,2023-4-8.
[5] 底盘运动学 全向轮系[EB/OL] .https://www.yltzdhbc.top/RobotSim-Omni,2022-4-29.
[6] 非 6020 舵轮开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22059,2022-7-31.
[7] RM2021-华南理工大学-普渡华南虎-舵轮步兵机器人开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12219,2021-8-25.
[8] 舵轮补档 [EB/OL] .https://www.bilibili.com/video/BV1nv4y117H3/?spm_id_from=333.880,2023-1-15.
[9] RM2021_深圳大学_RobotPilots 战队_自适应舵轮双枪步兵_机械开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12325,2021-8-31.
[10] RM2022-哈尔滨工程大学 Nooploop 创梦之翼-双枪平衡步兵机械开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=22138,2022-9-16.
[11] 底盘逆运动学解算[EB/OL] .https://zju-helloworld.github.io/Wiki/组件说明/机器人通用组件/算法/底盘逆运动学解算,2022-11-13.
[12] RM2021 - 华南理工大学 - 普渡华南虎 - 舵轮步兵电控开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12207,2021-8-24.
[13] RM2021 - 东北大学 - T-DT - 双枪步兵电控开源[EB/OL] .
https://bbs.robomaster.com/forum.php?mod=viewthread&tid=12290,2021-8-3.
[14] HW 自动步兵不完整形态视频[EB/OL] .
https://www.bilibili.com/video/BV1XT4y1a7bZ/?spm_id_from=333.337,2022-04-25.

[15] Lin J, Zhang F. R 3 LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package[C]. 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 10672-10678.

[16] Xu W, Cai Y, He D, et al. Fast-lio2: Fast direct lidar-inertial odometry[J]. IEEE Transactions on Robotics, 2022, 38(4): 2053-2073.

[17] Ren Y, Song X, Gao S. Research on Path Planning of Mobile Robot Based on Improved A* in Special Environment[C]. 2019 3rd International Symposium on Autonomous Systems (ISAS). IEEE, 2019:12-16.

[18] 王海芳,张瑶,朱亚锟等.基于改进双向 RRT*的移动机器人路径规划算法[J].东北大学学报(自然科学版),2021,42(08):1065-1070+1142.

[19] 朱冬,方向明,宋雯,唐国梅,张建. 基于改进势场蚁群算法的移动机器人全局路径规划方法[P]. 重庆市：CN115328148A,2022-11-11.

[20] Liu T, Yan R, Wei G, et al. Local path planning algorithm for blind-guiding robot based on improved DWA algorithm[C]. 2019 Chinese Control And Decision Conference (CCDC). IEEE, 2019: 6169- 6173.

[21] 刘二辉,姚锡凡,蓝宏宇,金鸿.基于改进遗传算法的自动导引小车动态路径规划及其实现[J].计算机集成制造系统,2018,24(06):1455-1467.DOI:10.13196/j.cims.2018.06.015.

[22] Park M G, Jeon J H, Lee M C. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing[C]. 2001 IEEE International Symposium on Industrial Electronics Proceedings. IEEE, 2001, 3: 1530-1535.

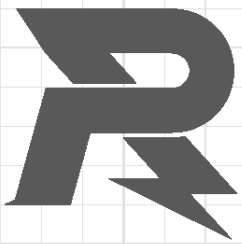
[23] 劳彩莲,李鹏,冯宇.基于改进 A*与 DWA 算法融合的温室机器人路径规划[J].农业机械学报,2021,52(01):14-22.

[24] Critical-HIT-hitsz/RMUA2022[EB/OL] . https://github.com/Critical-HIT-hitsz/RMUA2022,2023-2-8

[25] 李坤芝.导电滑环技术研究[J].舰船科学技术,1994(05):56-58.

[26] 杨莉红.导电滑环技术的研究[J].科技传播,2012,No.63(06):45+50.

[27] 刘世玺,李艳杰,张昊等.基于 DIC 的退火 NiTi 形状记忆合金力学性能研究[J/OL].热加工工艺:1-7[2023-04-06]



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