

# 浙狮战队 25 自瞄调参手册

## 前言

调参的过程也是深度掌握本自瞄程序的过程, 调试人员一定能在其中感受到本程序的局限性, 通过本调参手册, 旨在让调试人员能够快速掌握调参流程和手法, 借助调参加深对自瞄的学习。

测试对象需要能够以一定速度旋转和移动。转速区间为 0~130RPM, 场地上需要保证能够击打尽可能远的目标, 本调试手册以 5m 射程为例。

在调参流程结束后, 需要立即进行测试, 如果测试效果不佳, 则反过来再对调试参数进行优化, 并再进行测试, 以此循环。在测试达到预期后, 需要将最终测试的数据和视频进行整理。

调试者务必完成调参手册中的内容, 当然并不介意夹杂私货, 例如测试 130 转以及以上目标。

调试者尽可能自备 u 盘。

## 近赛期调参原则

对于非近赛期, 时间充裕, 建议在允许情况下调得越细越好, 该原则是在时间并不充裕, 不同兵种调试时间可能冲突导致需要较为快速调试的原则

1. 半小时原则: 对于任意需要调试的参数, 至多调试半小时, 不能拖延太多时间, 尽量将总调试时间缩短到 12 小时以内
2. 三分法原则: 如果想快速获取最优参数, 先取一个较大值, 再取一个较小值, 一直取中间值比较逼近最优解

## 调参前需要熟知的基本技能操作

1. MVS 使用(调整曝光并设定参数, 采样照片, 录屏)
2. VS 使用
3. Ros2 基本指令

## 基本调试方法

无 build 编译:

```
colcon build --packages-select auto_aim_interfaces
source install/setup.bash
colcon build
```

有 build 编译:(source ../install/setup.bash 可放入 ~/.bashrc 文件)

```
source install/setup.bash
colcon build
```

运行:  
ros2 launch auto\_aim\_bringup debug.py

建议更改 debug.py 中的参数文件(.yaml)名称,有利于保存.参数文件在“auto\_aim\_bringup/config”文件夹中,脚本文件在“auto\_aim\_bringup/launch”中

## 参数文件中的一些常用选项

# 选择射击模式 0:默认模式  
# 1:跟随模式 移动陀螺  
# 2:中心击打模式 移动陀螺  
# 3:精准击打模式 原地旋转  
# 4:平动击打模式  
# 5:定频射击模式  
# 6:持续射击模式  
# 7:取消射击模式,即挂机  
shoot\_state: 0: 调试不同参数需要对于的火控模式  
close\_shoot: false:在一些调试场景下不能启动自动开火  
show\_detect: false # 仅展示 detect 层  
show\_process: true # 仅展示 process 层  
vxz\_truncation\_rotate\_rpm: 120.0 # 转速过高截断值  
vxz\_truncation\_distance: 400.0 # 距离过远截断值  
在一些观测环境下需要截断 vx,vz,设置其为 0 即可,如果必须不进行截断,设置为 10000.0 即可

## 详细调参流程

各个兵种的调试模块

	步兵	英雄	哨兵
调整曝光	√	√	√
相机内外参	√	√	√
基于机械图纸获取坐标系转换数据	√	√	√
需要提前获取的其他数据	√	√	√
识别层	√	√	√
测试通讯	√	√	√
调试弹道偏移量和抬头补偿	√	√	√
调试朝向角	√	√	√
调试发弹延迟	√	√	√
调试中心击打	√	√	√
验证精准击打	√	×	√

调试模式切换参数与截断值	√	√	√
调试跟随击打	√	×	√
调试平移补偿	√	√	√
调试观测延迟	选做	选做	√
测试抬头补偿	√	√	√
原地陀螺击打测试	√	√	√
射程测试	√	√	√
另一色方击打测试	√	√	√
平移击打测试	√	√	√
移动陀螺击打测试	√	×	√
记录关键参数	√	√	√

### 1. 调整曝光

`exposure_time: 8000`

`gain: 32.0`

对于比较暗的环境,调整 `exposure_time=10000` 即可,而对于较亮的环境,可以降低 `gain`(增益),越高的曝光会降低帧率

### 2. 相机内外参

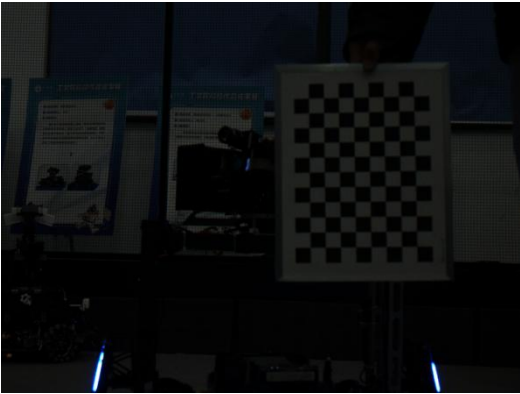
`intrinsic: [1680.865503159629, 0.0, 740.3637337990423,0.0, 1687.782004868199,`

`730.9768716652803,0.0, 0.0, 1.0] # 相机内参`

`distcoeffs: [0.01788831281527822, -0.2019999999070728, 0.004956905866419092,`

`-0.002866065261920429, 0.6039258624991095] # 相机外参`

打开 MVS,保证设定曝光,竖直放置相机标定板保证每个画面角落都掠过相机,总采样 40 张左右。样例:



Windows 食用方法:创建一个文件夹;放入 `bmp` 文件并将文件名称改为 `1.bmp` 到 `XX.bmp` 递增.之后在 VS 中 `camera_calibrate.cpp` 文件内更改 `dir` 的路径为文件夹路径并输入文件张数,运行,结束后文件 `calibration_result.txt` 即可获取相机内外参。

注意:如果相机参数不妥,图像中可能出现扭曲现象

### 3. 基于机械图纸获取坐标系转换数据

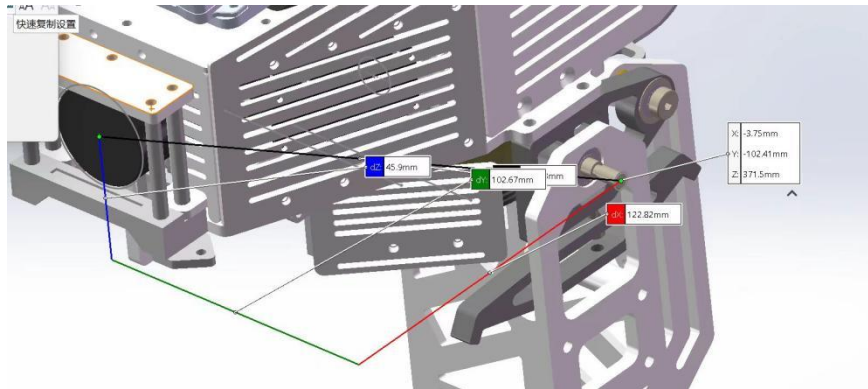
1. 基于机械图纸获取相机坐标系到绝对坐标系的位移 `xyz`,单位是 `cm`

`shaft_differ_x: 0.0 # 绝对坐标系 x 方向偏移量`

`shaft_differ_y: -4.59 # 绝对坐标系 y 方向偏移量`

`shaft_differ_z: 12.282 # 绝对坐标系 z 方向偏移量`

样例:画面中蓝色对应 y, 红色对应 z, x 方向即面朝相机的水平方向, 为 0



## 2. 基于机械图纸获取相机坐标系到射击坐标系(两摩擦轮中心)的位移 xyz, 单位是 cm

`sot_shaft_differ_x: 0.0` # 转到射击坐标系偏移 x

`sot_shaft_differ_y: -4.66` # 转到射击坐标系偏移 y

`sot_shaft_differ_z: 12.371` # 转到射击坐标系偏移 z

和绝对坐标系同理从相机面出发, 只是目标点为两个摩擦轮的圆心中点

## 4. 需要提前获取的其他数据

1. `bullet_speed: 2150.0` # 弹速(cm/s): 联系硬件与嵌入获得, 保证弹速在不超比赛要求上限的前提下, 尽可能高
2. (选做) 发弹延迟: 嵌入提前测试, 保证发弹延迟区间在 50ms 以内弹道散度尽可能低, 以散度不能超过一个装甲板为例, 尽可能在 5m 以内散度小于一个小装甲板
3. 是否双发: 机械问题, 联系机械, 尽可能不要双发
4. 射频(英雄不做): 击打固定装甲板, 计时后根据掉血和时间获取射频

## 5. 识别层

`yolo_model_path: "/home/nuc/projects/ruirui_love_v9/src/auto_aim/models/yolo.onnx"` # yolo 模型路径

`number_model_path: "/home/nuc/projects/ruirui_love_v9/src/auto_aim/models/fc2.onnx"` # number 模型路径

`use_fusion_detector: false` # 使用融合识别

`pic_min_threshold: 200.0` # 二值化参数

`armor_min_height_ratio: 0.6`

`small_armor_min_aspect_ratio: 0.8`

`small_armor_max_aspect_ratio: 3.2`

`big_armor_min_aspect_ratio: 3.2`

`big_armor_max_aspect_ratio: 5.0`

`max_center_angle: 15.0`

`number_min_confidence: 0.5` # 最小置信度

`num_img_roi_light_length: 12` # 灯条 roi 长度设置

保证模型路径正确, 否则运行会报错。设置 `show_detect=true`, `use_fusion_detector=true`, 保证识别颜色方正确。如果识别效果不佳, 调整曝光, 保证图像的清晰度, 如果识别依然有问题, 联系算法组组长。

设置 `use_fusion_detector=false`, 保证识别颜色方正确, 调试相关灯条匹配的二维尺寸参数, 还可以调试数字识别优化其鲁棒性, 保证任意转速下稳定识别。

**注意:** 调试者必须非常熟悉传统识别的调试流程, 保证比赛日神经网络如果出现问题, 来得及调试。

## 6. 测试通讯

`negation_send_yaw: false` # 对 `send_yaw` 取反

`negation_send_pitch: false` # 对 `send_pitch` 取反

`negation_read_yaw: false` # 对 `read_yaw` 取反

`negation_read_pitch: false` # 对 `read_pitch` 取反

1. 设置 `serial_read_data_node.debug=true`, 观察 `yaw`, `pitch` 是否随着时间在不断自增, 以及云台在平放下是否接近 0
2. 设置 `show_process=true`, 遥控器不开自瞄模式, 尝试上下摆动 `pitch`, 如果目标四装甲板状态畸形, 对 `read_pitch` 取反, 同理左右摆动 `yaw`, 如果四装甲板出现大范围摆动, 对 `read_yaw` 取反
3. 开启自瞄, 如果瞄准方向不正确, 对对应方向取反

## 7. 调试弹道偏移量和抬头补偿

`sot_shifting_yaw: -1.5` # 转到射击坐标系偏移 `yaw`

`sot_shifting_pitch: -4.0` # 转到射击坐标系偏移 `pitch`

`pitch_compensation_amplitude: 1.45` # `pitch` 补偿增强权重

设置 `show_detect=false`, `show_process=true`, `close_shoot=true`, `pitch_compensation_amplitude=0`, 关闭自动开火, 将静态装甲板正向镜头, 并接近到 1m 以内。调试偏移量至打弹可以命中中心。接着设置 `pitch_compensation_amplitude` 为 1.0, 开始让静态装甲板远离镜头, 逐渐微调 `pitch_compensation_amplitude`, 保证抬头补偿在射程以内能够基本命中车装甲板中心。

**注意:** 根据上赛季经验, 在上场前尽可能再检查一下抬头补偿, 似乎会因为时间发生偏移

## 8. 调试朝向角

`inclined: 0.392` # 倾斜性权重

`pixel_error: 1.8` # 倾斜性权重

`increase_param_blue: 2.6` # 蓝方朝向角增强参数

`increase_param_red: 2.6` # 红方朝向角增强参数

尝试调试倾斜性权重, 一般直接修改朝向角增强参数, 保证露出两个装甲板绝对值合为 90 度且装甲板转到中心时相对丝滑。此时将靶车改为另一个颜色, 调试对应的增强参数

## 9. 调试发弹延迟

`shoot_delay: 0.11` # 发弹延迟

设置 `svxz_truncation_rotate_rpm=0` (截断 `vx`, `vz`), `shoot_state=2` (中心击打模式),

`close_shoot=false`, 设置 `max_shoot_middle_x=1.5`, 2m 原地装甲板, 从 20rpm 逐步提升至至少 100rpm, 调试 `shoot_delay`, 保证能够尽可能命中装甲板中心。

## 10. 调试卡尔曼滤波器参数

`q_vs: [0.1, 0.1, 0.1, 0.1, 1.0, 0.001, 0.01, 0.85]` # 整车观测的观测矩阵

`r_vs: [0.1, 0.5, 1.0, 0.2]` # 整车观测的测量矩阵

`trans_q_vs: [0.01, 0.1, 0.01, 0.01, 0.01]` # 平动观测的观测矩阵

`trans_r_vs: [0.8, 0.5, 0.8]` # 平动观测的测量矩阵

1. `r_vs` 与 `trans_r_vs` 一般无需更改,参考 `default.yaml` 参数文件
2. `q_vs` 前四项(位置观测误差)设置越小,后四项(速度观测误差),速度拟合越快
3. 设置 `shoot_state=1`(跟随模式),2m 原地装甲板 30rpm,关闭遥控器,调试 `q_vs`,尝试用手晃动枪口,更可能保证 `vx,vz` 的晃动情况较稳定即可,调参者此步可不进行调试,即不更改 `q_vs`
4. 设置 `shoot_state=4`(平动击打模式),平移装甲板,调试 `trans_q_vs`,尝试用手晃动枪口,更可能保证 `vx,vz` 的晃动情况较稳定,启动自瞄不会发生过调,且满足拟合速度的需求。

## 11. 调试中心击打

`max_shoot_middle_x: 3.5` # 瞄准中心的射击坐标系下击打最小 x 尺寸

测试发弹延迟后,需要调试 `max_shoot_middle_x` 至合理值,2m 原地装甲板,转速设置为 80,尽可能提高 `max_shoot_middle_x`(一般为 3~4),观察击打效果,保证高射频

## 12. 验证精准击打

设置 `shoot_state=3`(精准击打模式),理论上应该对 2m,100rpm 以内的目标有 >80% 的命中率.如果效果不佳,尝试寻找问题

## 13. 调试模式切换参数与截断值

`min_shoot_track_rpm: 10.0` # 最小跟随击打 rpm(否则采用完全随动)

`min_shoot_middle_rpm: 60.0` # 最小瞄准中心 rpm

`min_shoot_precise_rpm: 80.0` # 最小精准击打 rpm

`vxz_truncation_rotate_rpm: 120.0` # 转速过高截断值

`vxz_truncation_distance: 400.0` # 距离过远截断值

1. 设置 `shoot_state=3`,`vxz_truncation_rotate_rpm=10000.0`,`vxz_truncation_distance=10000.0`,命中率高于 60% 下尽可能提高转速,极限转速即为 `vxz_truncation_rotate_rpm` 值,记录。
2. 同理设置 `vxz_truncation_rotate_rpm=10000.0`,`vxz_truncation_distance=10000.0`,测试低于极限转速 10rpm(参考,适当即可)的转速下,命中率高于 60% 下尽可能远距离,该极限距离即为 `vxz_truncation_distance` 值。最后记得将两个值全部赋予。
3. 正常情况下,直接让  
`max_shoot_middle_distance=vxz_truncation_distance`,`min_shoot_precise_rpm=vxz_truncation_rotate_rpm` 即可,保证精准击打在截断前提下运行,调试者也可以根据调试情况调整,但是一定要保证  
`max_shoot_middle_distance>=vxz_truncation_distance`,`min_shoot_precise_rpm>=vxz_truncation_rotate_rpm`; `min_shoot_track_rpm=10.0` 即可,在 10~20 之间基本可靠。

## 14. 调试跟随击打

`auto track_face_angle_func = [](const Eigen::VectorXf& car_state)` # 跟随击打朝向角区间映射函数

1. 设置 `shoot_state=1`(跟随模式),联系嵌入联调,调整控制参数,配合嵌入调试装甲板切换的控制参数;
2. 固定 2m,逐渐提高装甲板转速,每个不同转速和 *转向* 测试,调试 `track_face_angle_func`,调试直到击打效果不佳(参考低于 40%),此时的敌方转速 rpm 即为 `min_shoot_middle_rpm` 值。`rotate_control_compensation_left`,`rotate_control_compensation_right` 参考值为 0.05  
根据记录的不同转速下的最优 rpm,建立方程,放入“`auto_aim/node/ProcessorNode.cpp`”的  
`track_face_angle_func` 即可,

3. 使用 60~80rpm 自转,使用中心击打或精准击打,测试距离

**注意:**当前火控逻辑由于跟随击打在非切换情况下一直射击,所以跟随击打角必须调试足够精准

## 15. 调试平移补偿

`trans_control_compensation_left: 0.0` # 平移左转运动增强权值

`trans_control_compensation_right: 0.0` # 平移右转运动增强权值

1. 设置 `shoot_state=4`(平动击打模式), `close_shoot=false` 关火
2. 不进行旋转而是平移靶车, 逐渐提高移速并根据左右方向, 调试

`trans_control_compensation_left, trans_control_compensation_right`. 尽可能在低移速下能够锁住。

`trans_control_compensation_left, trans_control_compensation_right` 参考值为 `0.2`

## 16. 调试观测延迟

`auto delay_fuc = [](const int& kind_target)`

1. 由于观测器从 `0` 拟合需要一定的时间, 固然在拟合时间, 使敌方固定 `100rpm`, 设置 `show_process=true`, 用手机计时功能计时拟合到 `80rpm, 45rpm, 30rpm` 的时间, 分别对应于哨兵, 步兵, 英雄, 写入对应的 `delay_fuc` 函数中即可

注意: 至此, 参数已经全部调试完毕, 接下来是测试环节. 在测试前请确保以下参数正确

`set_enemy_color: 0` # 设置敌方颜色 请根据当前测试靶车选择, 测试结束后则需要设置为 `0`

`close_shoot: false` # 关火

`show: false` # 关闭显示图像

`shoot_state: 0` # 选择射击模式 `0`: 默认模式

# 详细测试流程

## 1. 测试抬头补偿

1. 装甲板正对枪管, 不进行旋转, 逐渐远离, 自动瞄准后人工打弹, 保证在射程内能够全部命中。
2. 将过程录制视频(共 1 个)

## 2. 原地陀螺击打测试

1. 2m 固定旋转, 分别在 `10rpm, 40rpm, 80rpm, 100rpm` 测试命中率
2. 将 2m 更改为最远射程, 分别在 `10rpm, 40rpm, 80rpm, 100rpm` 测试命中率
3. 将过程录制视频(共 8 个)

## 3. 射程测试

1. 将靶车调至 `70~80rpm`, 保证命中率大于 `50%`(英雄最好 `70%`, 哨兵可以再小一些), 此时的距离即为最远距离
2. 将最远距离时的击打情况录制视频

## 4. 平移击打测试

1. 将靶车在 2m 位置平移靶车, 速度从慢到快, 并保证自动开火
2. 将过程录制视频(共 1 个)

## 5. 移动陀螺击打测试

1. 保持转速为 `30rpm` 与 `60rpm`, 分别测试在射程内进行平移的效果, 这里的移动可以随意一些, 但尽可能在大部分情况下保持慢速和匀速。
2. 将过程录制视频(共 2 个), 建议测试者在移动陀螺击打这一项多进行测试

## 6. 另一色方击打测试

1. 在 2~3m 保持转速为 30rpm 与 60rpm, 测试击打效果
2. 如果测试结果不佳, 找组长反馈

## 7. 自身小陀螺状态下的自瞄效果






保持自身小陀螺, 注意安全, 2m 固定旋转, 分别在 40rpm, 80rpm 测试命中率。如果命中率影响明显, 联系兵种长

## 8. 记录关键参数

记录关键参数	样例
射程(m)	0.5~5
弹速(m/s)	21.5
发弹延迟(s)	0.65
射频(HZ)	21
全局平均命中率(%)	60%
全局平均 MDR	0.33
全局平均 DPS	70

1. 创建 excel 文档, 记录数据
2. 这里的射程包括最小射程, 调试者测试靶车在多近距离下自瞄失能即可。
3. 弹速, 发弹延迟, 射频之前均已测得, 直接填入
4. 基于原地陀螺击打和移动陀螺击打视频, 记录命中率和 DPS,  $MDR = DPS / \text{射频} * 10$ , 统计后取算术平均即可
5. 调试者也可分别对原地陀螺击打和移动陀螺击打各个击打视频统计的命中率和 DPS 记录入 excel
6. 如果有额外的测试项目, 可以单独记录。最后将 excel 文档和视频放在一个文件夹中压缩上传, 文件名称上清

楚易懂即可, 例如 (300 血, 2m, 10rpm):

 新步兵-2025-2-14-李瑞.xlsx	2025/2/14 0:44	XLSX 工作表	10 KB
 原地陀螺击打测试-300-2m-10rpm.mp4	2025/1/8 15:14	MP4 文件	13,967 KB
 原地陀螺击打测试-300-2m-40rpm.mp4	2025/1/8 15:16	MP4 文件	5,543 KB
 原地陀螺击打测试-300-2m-80rpm.mp4	2025/1/8 15:16	MP4 文件	14,784 KB
 原地陀螺击打测试-300-2m-100rpm.mp4	2025/1/8 15:16	MP4 文件	14,784 KB

注意: 测试者最好在测试结果良好并确定结束调试的前提下再记录关键参数, 以节省时间

## 9. 调试结束后的必查内容

以下参数必须设置

`set_enemy_color: 0` # 设置敌方颜色 0: 不设置 1: 敌方为蓝色 2: 敌方为红色

`close_shoot: false` # 关火

`show: false` # 关闭显示图像

`shoot_state: 0` # 选择射击模式 0: 默认模式

## 10. 快速自瞄自查攻略

1. 检查相机的两个螺丝拧紧情况, 确保均有螺丝固定
2. 检查自瞄小电脑的相机线是否稳定连接



3. 检查自瞄小电脑的串口线是否稳定连接
4. 使用装甲板图片,检查是否能够触发云台跟随
5. (算法组自查)检查以下参数是否设置正确:

`set_enemy_color: 0` # 设置敌方颜色 0: 不设置 1: 敌方为蓝色 2: 敌方为红色

`close_shoot: false` # 关火

`show: false` # 关闭显示图像

`shoot_state: 0` # 选择射击模式 0:默认模式

6. (算法组自查)接入显示屏查看曝光情况