

# （目标射击） 技术文档

参赛项目： 【算法应用赛】无人车任务挑战赛  
参赛队员： 刘洋、蒙俊章、韦家稳  
作品编号： **bdc23140392**  
参赛院校： 北部湾大学

## 目录

1. 背景介绍.....	2
2. 系统架构 .....	2
3. 传感器和执行器.....	3
4. 算法和控制 .....	5
5. 软件实现 .....	8
5.1 主要的 ROS 软件包.....	8
5.2 ROS 的调试和测试.....	10
6. 硬件实现 .....	10
7. 性能评估 .....	11
8. 总结和展望 .....	12
参考文献 .....	13

## 1. 背景介绍

全球校园人工智能算法精英大赛中的无人车目标射击挑战赛旨在推动机器人领域技术的创新和发展。该比赛设立了一系列任务，要求参赛机器人通过视觉感知、智能控制以及射击模块等关键技术，实现在虚拟或实际场景中准确识别和打击目标。比赛的目的是促使参赛者在人工智能、机器人技术、图像处理等领域展现卓越的创新能力，同时为培养未来人才提供一个实践和竞争的平台。

在这一挑战赛中，我们的机器人的角色是完成特定的射击任务，包括对移动或固定目标进行识别、瞄准和打击。机器人需要通过先进的人工智能算法和视觉识别技术，结合全向运动的机械设计，精确地执行射击动作。这些任务涉及到机器人在复杂环境中实现定位、导航和目标识别的综合能力，为参赛者提供了在实际应用场景中应对挑战的机会。整体而言，该竞赛旨在推动无人车技术的创新，促使参赛者深入理解并运用人工智能和机器人技术解决实际问题。



## 2. 系统架构

我们使用的机器人“ABOT-M1-ATK”人工智能无人车的系统架构设计旨在实现高效的目标射击任务，并结合 ROS 提供的强大功能来简化开发和集成过程。整体系统架构包括硬件和软件两个主要部分。

### （1）硬件部分：

机器人的硬件部分采用模块化设计，主要包括机械结构、电控系统、传感器和射击模块。机械结构采用高强度航空铝合金材质，具有模块化设计，方便扩展和维护。电控系统采用分布式控制，包括主控制器和运动控制系统。传感器方面，激光雷达、视觉传感器和九轴 IMU 模块等组成了全面的感知系统。射击模块由 ARM Cortex™-M4 内核控制，具备高速电机和电源系统，确保射击精准度和稳定性。

### （2）软件部分：

机器人的软件部分构建在 ROS 框架上，利用 ROS 的模块化和可重用性，实现了多个功能模块的协同工作。ROS 为机器人提供了一系列的工具和库，例如 RViz、MoveIt、Navigation 等，能够实现机器人通用的消息传递、设备驱动、导航规划等功能。具体软件实现包括：

● 人工智能控制：采用 INTEL 工控机，双核多线程，为机器人提供强大的计算能力，支持 ROS 和人工智能算法的运行。

● 视觉识别和目标跟踪：利用 1080P 高清摄像机进行图像采集，通过 ROS 中的视觉传感器模块实现人脸检测、目标跟踪等任务。

● 导航和运动控制：通过 ROS 中的导航模块，结合激光雷达和九轴 IMU 数据，实现机器人的定位、导航和全向运动控制。

● 射击控制：射击模块的控制基于 ARM Cortex™-M4 内核，通过 ROS 的通信机制实现与主控制器的协同工作，确保射击任务的准确执行。

ROS 在整个系统中起到了关键的中介角色，提高了系统的可维护性和灵活性。通过 ROS，不同功能模块可以独立开发和测试，然后集成到整个系统中，简化了开发流程，提高了系统的稳定性和可靠性。

### 3. 传感器和执行器

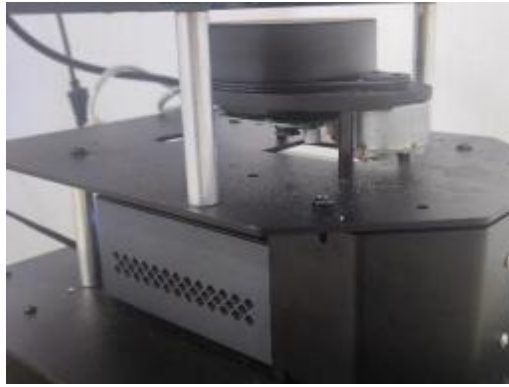
“ABOT-M1-ATK”人工智能无人车采用多种传感器和执行器，以实现全方位感知和高效执行目标射击任务。以下是主要的传感器和执行器以及它们在 ROS 系统中的集成情况：

#### （1）传感器：

● 激光雷达：机器人配备 360° 扫描的激光雷达，测距范围为 0.15-12m，测距分辨率为 0.5mm，频率为 8000Hz。激光雷达通过 ROS 中的激光雷达驱动包（如 `urg_node`）进行数据采集，并利用 `gmapping` 或其他 SLAM 算法进行地图构建和定位。

● 视觉传感器：采用 1080P 高清摄像机，帧率可达 120 帧/秒，视角为 120°。在 ROS 中，通过使用 `usb_cam` 或其他相应的摄像头驱动包，实现对图像的采集和处理，支持人脸检测、目标跟踪等任务。

● 九轴 IMU 模块：包括三轴陀螺仪、三轴加速度计和三轴磁场传感器，提供机器人的姿态信息。通过 ROS 中的 IMU 驱动包进行数据读取和处理，为导航和运动控制提供重要的信息。



(2) 执行器：

● 麦克纳姆轮：采用四个直径为 100mm 的铝合金麦克纳姆轮，通过四轮独立驱动实现全向运动。ROS 中的底盘控制模块可以与麦克纳姆轮相结合，实现机器人的灵活移动。



● 射击模块：射击模块包括主控制器（ARM Cortex™-M4 内核）、高速电机、电源系统等组件。在 ROS 系统中，通过定义射击控制节点，通过主控制器与 ROS 进行通信，实现射击任务的协调与执行。



传感器和执行器通过 ROS 的驱动和控制方法，实现了与机器人系统的良好集成。ROS 的节点通信机制和标准消息格式有助于简化传感器和执行器的接口，使得不同组件之间可以高效地交换信息，从而实现整体系统的协同工作。常见的 ROS 驱动程序包括：

**hokuyo\_node**：该驱动程序可以实现对 Hokuyo 激光雷达的采集和数据发布，并且提供了多种参数设置和滤波选项。

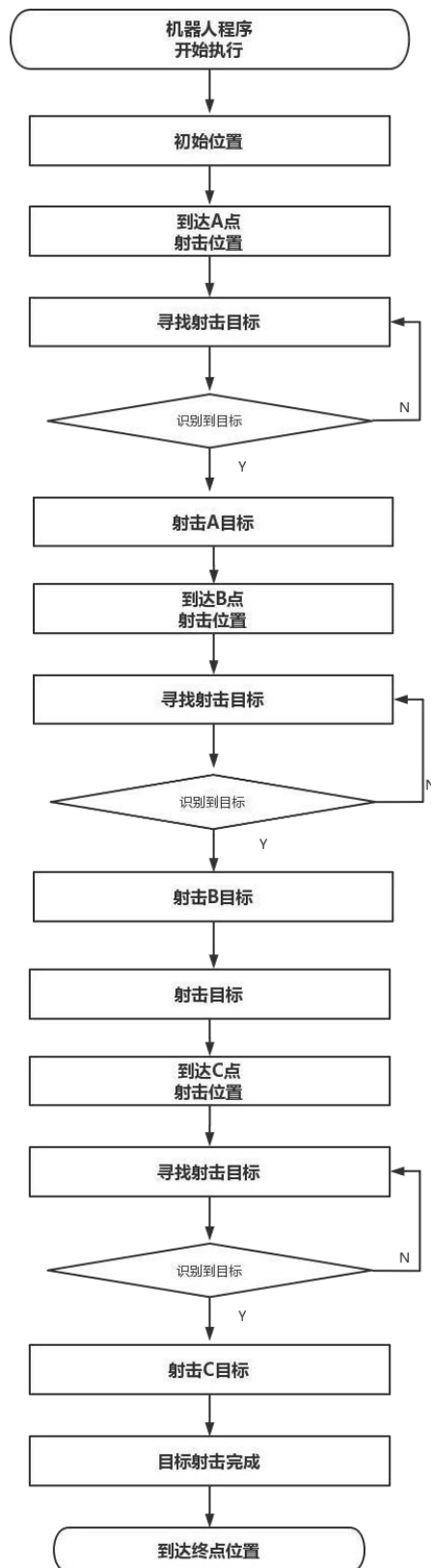
**usb\_cam**：该驱动程序可以实现对 USB 摄像头的采集和数据发布，并且支持多种分辨率和帧率选择。

**rosserial**：该驱动程序可以实现对 Arduino 等嵌入式设备的 ROS 通信支持，并且提供了多种数据类型和消息格式的转换方法。

**Pololu Jrk Motor Controller 驱动程序**：该驱动程序可以实现对 Pololu 电机驱动器 的控制和参数设置，并且支持PID 闭环控制。

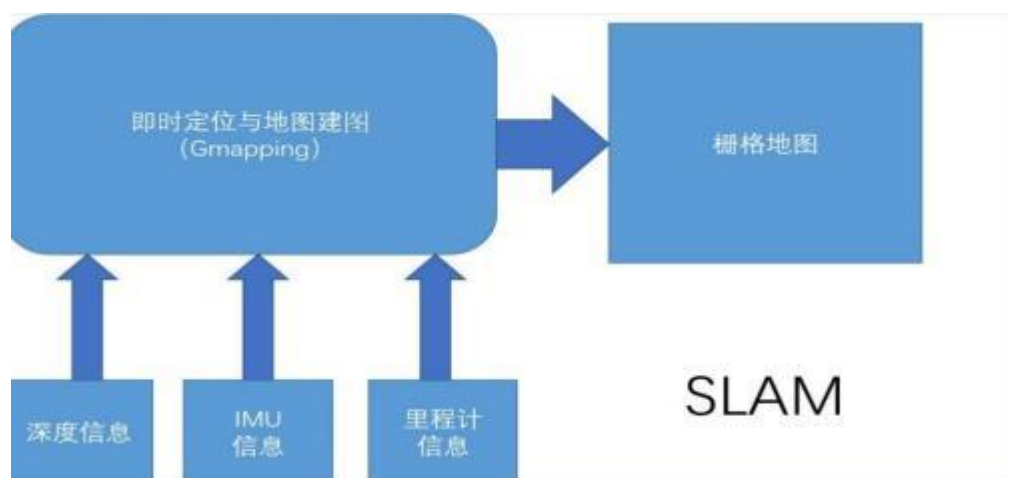
## 4. 算法和控制

目标射击机器人无人小车通过激光雷达传感器对比赛场地进行全局扫描，获取详细的地图信息，包括障碍物和导航路径。这一步为后续的自主导航提供基础数据。获取的地图数据被保存至系统，同时启动AMCL（自适应蒙特卡洛定位）系统，通过激光雷达的实时感知，在保存的地图中进行自主定位。这确保了机器人在比赛场地中了解自己的准确位置。随后，小车在导航过程中持续使用激光雷达进行实时环境感知，并通过与保存好的全局地图对比，实现对自身位置的精确确认，从而实现高效的自主导航。一旦小车达到目标位置，启动视觉传感器对射击目标靶上的对应二维码进行识别，提供目标的具体位置信息。通过视觉跟踪算法，实时锁定射击目标的位置。最后，小车启动射击模块，在精确锁定目标的情况下进行射击，实现了射击打靶的任务目标。整个流程通过协同工作的传感器和执行器，以及ROS系统的集成，确保了目标射击机器人在比赛中高效、准确地完成任务。流程图如下图所示。



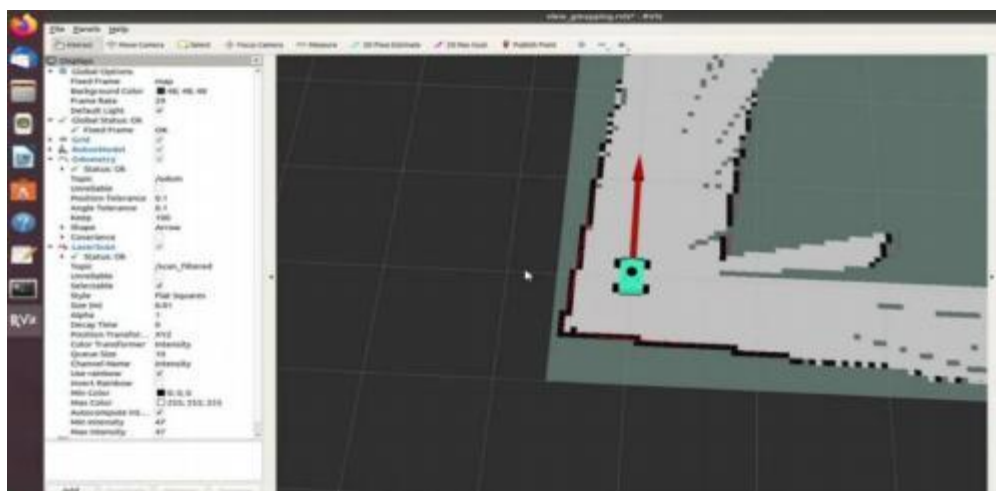
机器人无人车采用多种算法和控制方法， 用来实现在目标射击挑战赛中的任务。以下是主要的算法和控制方法以及它们在 ROS 系统中的实现方式：

(1) SLAM 算法：为了实现精确定位和地图构建，机器人采用了 ROS 中的 SLAM 算法。常用的 SLAM 算法包括 gmapping、hector\_slam 和 cartographer 等。通过激光雷达和视觉传感器提供的数据，机器人能够在未知环境中实时构建地图，并准确定位自身的位置。



SLAM算法

对于 SLAM 功能的调试，采用了 Gmapping 算法进行地图的建设与保存。通过手动控制机器人，使用激光雷达与里程计获取全局的先验地图，并将地图保存到本地。接着，依据 AMCL 进行自主定位，通过当前激光雷达与里程计对比已保存的全局地图，确定机器人的位置，并实现导航和避障。SLAM 算法建图如下：



(2) 路径规划和避障：机器人的路径规划和避障算法采用了 ROS 导航堆栈中的相关模块。利用 SLAM 构建的地图，机器人可以通过全局路径规划确定目标点，并通过局部路径规划在复杂环境中实现安全导航。避障功能通过激光雷达实时检测周围障碍物，保证机器人在移动过程中避免碰撞。



（3）姿态控制：九轴 IMU 模块提供了机器人的姿态信息，通过 ROS 中的控制节点，可以实现对机器人姿态的精准控制。这对于确保机器人在目标射击过程中能够保持稳定的姿态，提高射击精度至关重要。

（4）射击控制：射击模块的控制采用了 ARM Cortex™-M4 内核，通过 ROS 中定义的射击控制节点进行协调。这个节点负责接收目标信息，计算射击方向和角度，并控制高速电机实现射击动作。整个过程通过 ROS 消息传递实现，确保与其他系统模块的协同工作。

在实现目标射击挑战赛任务的过程中，机器人使用了多层次的算法和控制策略，确保了整体系统的协同运行。在路径规划方面，ROS 导航堆栈不仅提供了全局路径规划，还允许机器人根据实时感知的环境信息进行局部路径规划，使得机器人能够在狭窄或动态变化的环境中灵活应对。避障算法通过激光雷达实时监测障碍物，采用先进的避障策略，使得机器人在高速移动中能够快速、安全地避免碰撞。

姿态控制模块通过 ROS 节点实时接收和处理九轴 IMU 模块的数据，确保机器人能够稳定运动，有效提高了射击的准确性。而射击控制方面，ARM Cortex™-M4 内核的射击模块通过 ROS 消息传递协同工作，将目标信息转化为精准的射击动作，保证了在高速移动中仍能精确锁定目标。

这些算法和控制方法的综合运用使得“ABOT-M1-ATK”无人车在目标射击挑战赛中表现出色。ROS 作为开发平台的核心，为系统整合提供了灵活性和可扩展性，使得机器人能够适应不同任务场景的需求。整体而言，机器人在该挑战赛中的表现不仅仅是技术实力的展示，更是对多领域知识和算法综合运用的体现，为未来无人车和人工智能领域的研究提供了有力支持。

## 5. 软件实现

### 5.1 主要的ROS软件包

软件包	组成	功能	使用方法
robot_shoot	包含了与机器人射击模块相关的功能和节点。	该软件包负责射击功能，包括对射击模块的控制、射击算法的实现以及与导航和感知模块的集成，确保机器人能够准确锁定并射击目标	通过roslaunch robot_shoot nav_shoot.launch等命令启动，其中 nav_shoot.launch是射击模块的启动文件。

abot_bringup	包含了机器人初始化和启动的相关配置，其中包括传感器的启动、IMU的配置等	这个软件包负责机器人的启动和初始化。它包括对传感器和执行器的配置，确保系统在启动时处于正常工作状态。此软件包还可以处理一些启动时的自检和系统检测任务	通过roslaunch abot_bringup robot_with_imu.launch 命令启动，其中 robot_with_imu.launch 是一个启动文件，包含了机器人初始化的相关配置。
abot_imu	包含了与IMU相关的功能和节点	IMU（惯性测量单元）在机器人导航和姿态控制中起着关键作用 abot_imu软件包涵盖了对IMU传感器的驱动和数据处理，以获取关于机器人姿态和角速度的信息	与其他节点协同工作，通过ROS话题或服务进行数据交换。
abot_model	包含了机器人的模型描述和相关配置	abot_model软件包包含机器人的物理特性、动力学模型等信息，以便在仿真环境中准确模拟机器人的行为	在机器人控制和导航节点中引用模型信息
abot_tool_pkg	包含了一些机器人工具和辅助功能	用于调试、监测和执行其他开发任务。还包括对节点性能的监测、日志记录工具等	可以根据需要引用这些工具来帮助调试机器人系统
lidar_filters	包含了与激光雷达数据处理相关的功能和节点	lidar_filters软件包提供了对激光雷达数据进行滤波和预处理的功能，以提高数据的质量。	通过ROS话题接收激光雷达数据，输出处理后的数据供其他节点使用
robot_slam	包含了与SLAM相关的功能和节点	robot_slam软件包负责实现SLAM功能，通过激光雷达等传感器数据生成地图并实现机器人在地图中的定位	通过roslaunch robot_slam_navigation.launch等命令启动，其中 navigation.launch是SLAM的启动文件。
track_tag	包含了与目标跟踪相关的功能和节点	track_tag软件包涵盖了对目标标签的检测、识别和跟踪算法，以确保机器人能够锁定射击目标的位置	通过roslaunch track_tag_usb_cam_with_calibration.launch等命令启动，其中 usb_cam_with_calibration

			.launch是启动相机和标定的配置文件
--	--	--	----------------------

机器人的软件实现基于 ROS，这为系统提供了强大的开发框架和工具。ROS 软件包的组成涵盖了多个关键领域，包括感知、控制、导航和射击等功能模块。这些ROS软件包的紧密协作构成了机器人系统的全面功能,通过roslaunch命令启动各个节点，实现对机器人系统的灵活调度。abot\_bringup提供了机器人的基础启动和初始化，abot\_imu处理IMU数据，abot\_model描述机器人的物理模型，abot\_tool\_pkg提供辅助工具，lidar\_filters进行激光雷达数据处理，robot\_shoot控制射击模块，robot\_slam负责SLAM，而track\_tag处理目标跟踪。这些组件协同工作，使得机器人能够在目标射击竞赛中完成从环境感知到决策执行的全流程任务。

## 5.2 ROS的调试和测试

在ROS中，调试和测试是确保机器人系统稳定性和性能的关键步骤。为了有效地进行调试和测试，我们采用下面方法进行。

首先，监视ROS节点的输出是一种常见的调试方法。通过查看节点的终端输出，可以了解节点是否正常运行，以及是否产生了预期的结果。例如，可以使用rostopic echo和rostopic hz命令监视话题的消息发布和频率，以确保传感器数据正常接收。其次，检查传感器数据的正确性也是调试的一个重要方面。通过使用RViz或rostopic echo等工具，可以实时查看传感器数据，确保其与实际场景一致。在目标射击竞赛中，激光雷达和视觉传感器的准确性对于正确锁定目标至关重要。此外，ROS提供了一系列的调试工具，如rqt、roslaunch rqt\_graph rqt\_graph、roslaunch rosbag等，这些工具可以用于可视化地监视ROS系统的运行状态、节点之间的通信关系，并记录和回放节点的消息，有助于排查问题和优化性能。

对于测试方法，使用rostopic工具，可以方便地进行自动化测试，检查系统在不同条件下的表现。这对于保证机器人在目标射击竞赛中的稳定性和可靠性至关重要。通过有效利用ROS提供的工具和命令，结合对传感器数据和节点输出的监视，可以高效地进行调试和测试，确保机器人系统在目标射击竞赛中能够如期表现。

## 6. 硬件实现

机器人的硬件实现涵盖了机械结构、电路连接和电源管理等关键方面，确保了系统的稳定性和可靠性。

（1）机械结构方面，采用了高强度航空铝合金材质的车体骨架和外壳设计，使机器人具备轻量化和强度优越的特点。机械结构采用了模块化设计，便于后续的扩展和维护。四个直径为 100mm 的铝合金麦克纳姆轮的布置使得机器人能够实现全向运动，提高了机动性和精度。

（2）电路连接方面，我们采用分布式控制系统，主控制器搭载了Intel 双核处理器，用于运行 ROS 系统和控制机器人的整体运动。电机驱动系统采用矢量控制和 PWM 技术，以

实现对四个工业级直流有刷伺服电机的精准控制。同时，激光雷达、视觉传感器等各类传感器通过多路接口与主控制器相连接，实现对外界环境的感知。

（3）电源管理方面，机器人搭载了 12V10Ah 动力锂电池组，采用进口电芯，具备过充、过放、过流、短路保护等多重保护机制，确保电源系统的稳定性和安全性。射击模块独立供电，具备 8.4V4400mAh 的电池，通过电量显示模块实时监测电量，确保在目标射击挑战赛中持续稳定的射击性能。

ROS 和其他软硬件之间的接口采用了常见的通信方式，包括 RS-232、USB、CAN、蓝牙和 WIFI。这些接口保障了不同硬件模块之间的有效通信，确保整个系统能够高效地协同工作。机器人硬件实现的结构设计合理，电路连接清晰，为机器人在目标射击挑战赛中的稳定运行提供了可靠的硬件支持。

## 7. 性能评估

为了评估机器人的性能，我们将进行一系列的测试，包括速度测试、精度测试、障碍物避让测试等。通过模拟比赛场景和实地测试，我们将收集并分析机器人在不同情境下的表现。同时，我们将根据比赛规则和得分标准，对机器人在比赛中的实际表现进行评价。性能优化方面，我们将根据测试结果对机器人的控制算法、传感器参数进行调整，以提高其在比赛中的表现。通过不断的优化和调试，确保机器人在挑战中达到最佳状态，提高比赛中的竞争力。

### （1）速度评估：

**移动速度：**这一方面的评估涉及到在平坦地面上进行直线行驶测试。通过记录机器人在单位时间内能够行驶的距离，并计算平均移动速度，可以获取对机器人整体移动性能的评估

**执行速度：**机器人执行任务的速度是关键指标。通过测试机器人完成特定任务所需的时间，例如从起点移动到终点并完成目标任务，可以评估其执行速度。

**响应速度：**机器人对外界刺激的响应速度也是关键因素。测试机器人对命令或传感器的响应时间，以评估其对外部环境变化的快速适应能力。

### （2）精度评估：

**定位精度：**在已知环境中进行自主导航测试，记录机器人到达目标位置的误差范围。这一测试可以帮助评估机器人的定位系统的准确性和精度。

**感知精度：**评估机器人在感知环境中的准确性，包括目标识别、避障和导航。通过测试机器人在各种环境条件下的感知能力，可以获取其在复杂场景中的性能表现。

### （3）可靠性评估：

**稳定性：**机器人在不同环境条件下的运行状况是评估其可靠性的重要指标。通过测试机器人在崎岖路面或不同温度条件下的稳定性，可以了解其在各种工作环境中的表现。

**故障率：**连续运行一段时间后检查机器人的故障发生情况，评估其在长时间运行中的稳定性和可靠性。

**维护性：**测试机器人在出现故障后的维修和保养情况。包括维修时间和维修难易程度的评估，对机器人的整体可靠性提供重要信息。

为了对机器人进行性能测试和优化，采取了以下方法：

（1）构建多样化测试场景：通过构建多种测试场景，包括平坦路面、崎岖路面、室内和室外环境等，以全面评估机器人在不同工作场景中的性能。

（2）自动化测试脚本：编写自动化测试脚本，能够在各种场景中自动运行机器人并记录性能数据。这样的脚本可以生成详细的测试报告，为性能分析提供便利。

（3）数据分析：对测试数据进行深入分析，找出机器人性能的瓶颈和不足之处。通过对比不同测试方案的数据表现，选择更优参数或算法进行性能优化。

（4）硬件升级：针对发现的硬件瓶颈，进行硬件设备的升级。例如，更换更高性能的电机电、增加传感器精度等，以提高机器人整体性能。

（5）软件优化：对ROS软件包进行优化，包括提高消息传递效率、减少不必要的计算开销等。这些优化可以提高机器人在执行任务时的效率。

通过这些方法的综合运用，对机器人进行了全面的性能测试和优化，提高了其速度、精度和可靠性等方面的性能表现。这些工作对于确保机器人在实际应用中表现稳定、高效具有重要意义。

## 8. 总结和展望

参与机器人目标射击挑战赛是一次富有挑战性和收获的经历。通过比赛，我们不仅成功完成了比赛规定的任务，还在技术和团队协作方面取得了一系列进步。在比赛过程中，我们遇到了各种技术难题和挑战，例如，在调试阶段，对目标点 A、B、C 的参数进行精准调整是一个较为繁琐的过程，需要不断优化机器人的运动轨迹。此外，射击标靶的精度和准确性也是一个需要不断提升的技术难点。从经验和教训中，我们深刻认识到对待团队合作的重视和沟通的必要性。在团队中，每个成员都发挥了自己的优势，共同克服了技术难题。在面对挑战时，团队的凝聚力和协作精神是取得成功的关键。通过团队的共同努力，成功地克服了这些困难，最终实现了高效可靠的机器人竞赛方案。

BOT-M1-ATK 作为“无人车目标射击挑战赛”的机器人，通过高精度的传感器、强大的人工智能控制器以及灵活的运动系统，具备了在复杂比赛场景中执行任务的能力。全向运动、精准射击、激光 SLAM 等功能的融合使其在比赛中有着显著的竞争优势。在硬件实现



方面，ABOT-M1-ATK 采用高强度航空铝合金结构，麦克纳姆轮的应用提高了运动的效率和精度。搭载的人工智能控制器、激光雷达、1080P 高清摄像机等传感器以及射击模块的 ARM Cortex™-M4 内核，为机器人提供了强大的感知和执行能力。在软件实现方面，ROS 系统的运用使得机器人的软件系统更具模块化和可维护性，同时提供了丰富的功能包和开源代码，为开发者提供了广阔的创新空间。通过 ROS 的调试和测试方法，我们能够更加高效地验证和优化机器人的算法和控制策略。

展望未来，我们将继续深化对 ROS 框架的应用，进一步优化机器人的感知和决策能力。同时，我们计划在目标识别和射击方面探索更先进的技术，以提高机器人在复杂环境中的自主性和准确性。通过不断学习和创新，我们期待能够在未来的机器人竞赛中取得更加显著的成绩，并为机器人技术的发展做出更大的贡献。

## 参考文献

- [1] 黄昱洲;王立松;秦小麟. 一种基于深度强化学习的无人小车双层路径规划方法.计算机 2023-01-15
- [2] 陈高远. 无人小车规避路障与路径优化算法研究. 中国民航大学2021-05-28
- [3] 巩海军. 无人车视觉目标检测和路径规划算法研究. 北京交通大学2019-06-01
- [4] 谭浩 ;阳清;李江婷 ;张嘉麟. 基于 ROS 平台的巡逻车自主导航设计. 自动化与仪表2023-11-25
- [5] 胡晓强, 刘汉忠, 贾良冠等. 基于 ROS 平台的激光雷达 Slam-GMapping 的重定位方法研究
- [6] 陈锦儒, 林洁彬, 张成. 基于ROS系统的自主导航机器人设计 [J]. 电脑知识与技术, 2023, 19 (23): 99-101+105.
- [7] 汪杰. 多机器人协同SLAM算法研究[D]. 西安理工大学, 2023
- [8] 张军, 余志强. 基于ROS的室内巡检机器人多目标点导航研究与仿真 [J]. 绥化学院学报, 2023, 43 (06): 141-144.
- [9] 张小龙. 室内移动机器人SLAM与路径规划研究[D]. 南京信息工程大学, 2023.