

**调研报告**

课题名称 两轮自主机器人调研报告

学 院 电气工程学院

专 业 控制工程

班 级 194班

学 号 1912392039

姓 名 张溢炉

指导老师 李国进

二〇一九年十二月

1. 调研目的

了解两轮自主机器人的研究现状，并将相关技术应用到实验室的两轮自主机器人。

1. 调研方法

通过书籍、论文、视频（网上课程和讲座）、博客等途径进行调研。最后的参考文献是调研过程中阅读的文书籍和论文。

1. 调研结果

近些年，媒体经常报道由于高端技术机器人和人工智能而将发生变化的社 会、经济和文化的新闻。 随着社会发展，有些人以乐观的态度期待着不断提高改善的生活，但是可能会更早到来的对于劳动市场的负面影响的展望会使得人们更加不安。如此，我们身边正在进行的机器人和人工智能的研究和发展，会在不久的将来会对我们产生深远的影响。 所以我们更要关注机器人技术的发展，试着去了解并为未来做好准备。

机器人包含着很多技术要素，因此需要多方面的专业技术。实际上，机器 人为了要进入到我们的生活当中且被广泛运用，目前尚存在着很多技术局限性，需要进行更多的研究。想要克服当前的问题，专家、相关企业、一般用户需要携手努力一起发展现状。除了机器人的制作和应用之外，我们需要一个协作和开发的平台，我认为这就是ROS(Robot Operating System)平台。 ROS具备着降低技术壁垒和有益于传播的各种因素。 通过ROS平台，希望积累更多的知识和技术，使得更新更进步的机器人加入到

我们的生活中。

最近，“平台”在机器人领域也备受关注。平台分为软件平台和硬件平台。 机器人

软件平台不仅包括机器人应用中使用的硬件抽象、子设备控制，以及机器人工程中常

用的传感、识别、实时自定位和绘图（SLAM，Simultaneous Localization and Mapping）、导航（Navigation）和机械臂控制(Manipulation）等功能的实现，还包含功能包管理、开发环境所需的库、多种开发/调试工具。机器人硬件平台不仅包括移动机器人、无人机和人形硬件研究平台，还包括正在商业化的诸如SoftBank的Pepper和MIT Media Lab的Jibo等产品。

通过调研，最后决定以ROS中turtlebot3作为调研对象，研究其相关技术,并将其应用到自己的机器人上。以下是调研后对软硬件的总结。

1. 整体框架

机器人系统的整体框架如图3-1。机器人系统可分为硬件平台和软件平台。本次主要调研应用组件中视觉相关、地图构建、定位导航、路径规划等软件部分，包括SLAM和导航算法，SLAM分为视觉SLAM（简称VSLAM）和激光SLAM。其中，视觉SLAM是自己的主要研究方向。



图3- 1 机器人系统图

机器人的硬件种类很多，在ROS（Robot Operating System）出现之前，各硬件厂商没有统一的接口，广泛存在重复造轮子的现象。2007年11月，ROS诞生于斯坦福大学的STAIR项目，2010年ROS1发布，其最初目标是在机器人领域提高代码的复用率，解决重复造轮子的问题。在过去的十几年里，ROS社区中的功能包呈指数级增长目前已成为机器人领域的事实标准。ROS已成为了图3-1所示机器人系统通用的软件平台，其将整个系统模块化，并提供了统一的接口。很多硬件厂商也提供了相应的驱动功能包，无需再根据用户手册编写代码，只需安装和调用相应功能包。ROS使机器人设计和制作像搭积木一样，每个人只需完善自己擅长的部分。

ROS最初设计的目标机器人是PR2，这款机器人搭载了当时最先进的移动计算平台，网络性能优异，无需考虑实时性方面的问题，主要应用于科研领域。如今ROS应用的机器人领域越来越广：轮式机器人、人形机器人、工业机械手、室外机器人（如无人驾驶汽车）、无人飞行器、救援机器人等，美国NASA甚至考虑使用ROS开发火星探测器。机器人已经开始从科研领域走向人们的日常生活。ROS虽然仍是机器人领域的开发利器，但介于最初设计时的局限性，也逐渐暴露出了不少问题。

ROS1主要存在多机器人系统、跨平台、实时性、网络连接 、产品化等方面的问题，虽然很多开发者或者开发机构对其中一些问题提出了针对性的解决方案，但仍然无法解决ROS1中的根本问题。于是，ROS2应运而生，2017年12月8日ROS2发布第一个正式版本—Ardent Apalone。相比ROS1，ROS2的设计目标更加丰富，基本解决了ROS1中存在的问题，其支持多机器人系统、铲除原型与产品之间的鸿沟、支持微控制器、支持实时控制、支持跨系统平台。ROS2的关键中间件是DDS(Data Distribution Service，数据分发服务)，其最早应用与美国海军，用于解决舰船复杂环境中大量软件升级兼容性问题，目前已经成为美国国防部的强制标准，同时广泛应用于国防、民航、工业控制领域，成为分布式实时系统中数据发布/订阅的标准解决方案，实时性和安全性高。另外，ROS2使用了ROS1编译系统catkin的改进版ament。

虽然ROS2克服了ROS1的很多缺点，但在各平台的使用中还存在很多问题需要解决，功能包、教程也不及ROS1完善。对于初学这来说，ROS1可能是更好的选择，在熟悉ROS1后，可以将其迁移到ROS2。故接下来的调研将围绕ROS1展开。

ROS是中间件/类操作系统，包括硬件抽象、底层设备控制、常用函数实现、进程间消息传递、包管理，由通信机制或框架（分布式、进程管理、进程间通信）、开发工具（仿真、数据可视化、图形界面、数据记录）、应用功能（控制、规划、视觉、建图）、生态系统（ROS官网（邮件列表、answer、wiki）、社区等，软件包管理、文档、教程）四部分组成。其优点是节点容错性强、不同语言模块隔离、模块开发低耦合，缺点是过度依赖master节点（增加master镜像）、node节点异常问题、计算资源浪费、消息数据未加密。

ROS常用的开发环境是Roboware和Eclipse。ROS提供的有限状态机功能包SMACH可以使代码的逻辑更清晰，结构也会特别规整。另外，ROS还有丰富的建模和仿真组件，可以进行几何、运动学、动力学和控制学建模和仿真，可以通过仿真减少机械磨损，运行强化学习等算法，大大减小无人机调试时坠毁的数量。以下是ROS中常用的一些组件,一些组件还可以通过plugin拓展其功能。

* URDF（Unified Robot Description Format，统一机器人描述格式）：编程式建模，是描述机器人模型的XML格式文件，ROS提供URDF的C++解析器。
* xacro：一种特殊的URDF，提供了更高级的方式来组织编辑机器人描述，优化模型代码（通过宏定义复用代码）、提供编程接口，是URDF精简化、可复用、模块化的描述形式。
* sw2urdf：solidworks模型通过插件自动生成URDF模型。
* rviz：可视化显示URDF模型、显示所有检测信息、通过滑动条、数值等方法控制机器人。
* rqt：可视化调试和显示。包括日志输出工具、计算图可视化工具数据绘图工具、参数动态配置工具。
* gazebo：动力学建模，模型完善，各种传感器模型。模型（几何模型）与rviz模型相同，但是需要在模型中加入机器人和周围环境的物理属性，例如质量、摩擦系数、弹性系数等。
* airsim：适合无人机、高度还原真实场景，方便视觉的强化学习训练。
* arbotix：是一款控制电机、舵机的控制板，rviz+arbotix的传感器无法获取环境任何数据。
* ros\_contorl：控制学，提供了机器人控制中间件，封装了多种类型的控制器接口，传动装置接口、硬件接口、控制工具箱等，统一数据通信接口，多机器人硬件资源进行了抽象。
* ros+matlab+V-REP：MATLAB（2013之后）的robotics system toolbox提供了ROS的大部分功能，可以与ros通信，matlab可以结合V-REP进行仿真。
* rosbag：数据记录与回放，如记录摄像头或激光走完一条轨迹的数据之后，可以通过回放来仿真，而无需重复走同样的轨迹。

为了将ROS更好的应用于自己的机器人，我选择了TurtleBot3作为调研对象。TurtleBot3是一款两轮自主机器人，Waffle Pi版实物如图3-2，树莓派（Raspberry Pi）与远程PC之间使用无线通信。TurtleBot是ROS中最为重要的机器人之一，伴随ROS一同成长。作为ROS开发前沿的机器人，几乎每个版本的ROS测试都会以TurtleBot为主，ROS2也率先在TurtleBot上进行大量测试。因此，TurtleBot3是ROS支持度最好的机器人之一，可以在ROS社区中获得大量关于TurtleBot的相关资源，很多功能包也能直接复用到我们自己的移动机器人平台上，是使用ROS开发移动机器人的重要资源。

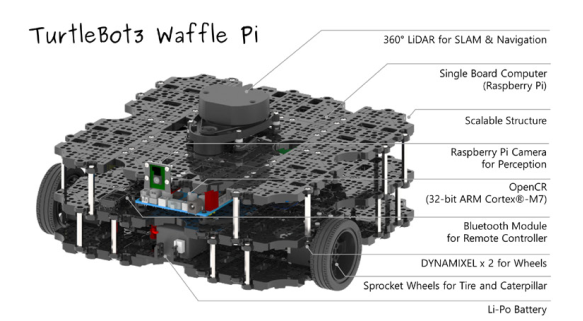


图3- 2 TurtleBot3 Waffle Pi实物图

TurtleBot第一代发布于2010年，两年后发布了第二代产品。前两代TurtleBot使用iRobot的机器人作为底盘，在底盘上可以装载激光雷达、Kinect等传感器，使用PC搭载基于ROS的控制系统。在2016年的ROSCon上，韩国机器人公司Robotis和开源机器人基金会（OSRF）发布了TurtleBot3，彻底颠覆了原有TurtleBot的外形设计，成本进一步降低，模块化更强，而且可以根据开发者的需求自由改装，提出了一种更加灵活的移动机器人平台。

TurtleBot3在远程PC上运行远程控制、SLAM和导航功能包；树莓派用于采集激光雷达和深度摄像头数据，并与远程PC通信；OpenCR是车上单片机，用于控制电机和采集一些传感器（如IMU）数据。

通过调研发现，安装于Ubuntu 16.04上的ROS Kinetic是目前应用最广泛的，同时Ubuntu 16.04也比较稳定。于是，我选择了ROS Kinetic来搭建自己的机器人。在开始学习ROS之前还必须先掌握Ubuntu、C++语言以及Cmake的使用，主要参考文献[3]-[4]。

1. 硬件平台
2. 软件平台
3. 视觉slam
4. 激光slam
5. 导航
6. 相关人工智能技术（EQA等，有篇周报有写）
7. 应用方案

参考文献

1. 表允晳,赵汉哲,郑黎蝹,林泰勋.ROS机器人编程[M]. ROBOTIS Co., Ltd.2017.12.
2. 胡春旭.ROS机器人开发实践[M].机械工业出版社.2018.05.
3. 鸟哥.鸟哥Linux私房菜（第三版）[M].人民邮电出版社.2010.07.
4. Stanley B.Lippman.C++ Primer(中文第5版).电子工业出版社.2013.09.
5. 《概率机器人》
6. Magnus Lie Hetland.Python基础教程（第3版）. 人民邮电出版社.2018.02

