

**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、社区、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。

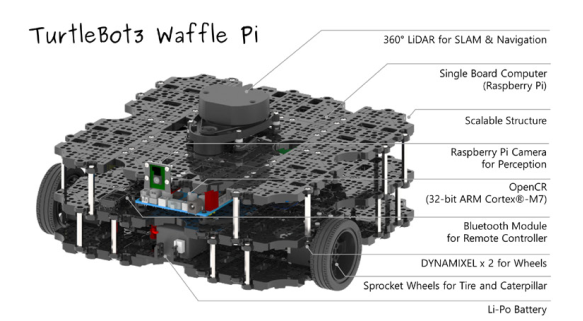


图3- 2 TurtleBot3 Waffle Pi实物图

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]。

以上是对整体框架的总结，接下来将分别介绍硬件平台和软件平台。

1. **硬件平台**

硬件平台可以分为控制层和执行层。我认为控制层可以按计算能力和体积大致分成三个梯队。第一梯队是云大脑、PC和工控机等，第二梯队是树莓派、Jetson Nano开发板和Jetson TX2开发板等，第三梯队就是微处理器，如STM32单片机、FPGA、DSP、Arduino等。执行层为图3-1中的驱动装置和感知系统。常见的驱动装置是电机，电机可大致分为伺服电机、舵机和步进电机。伺服电机一般给定速度控制信号，其可以一直朝固定方向旋转。舵机一般给定位置信号，其转动范围一般不超过360°。步进电机一般给定脉冲信号，电机根据脉冲的个数旋转相应的步数，如步进推杆。常用的传感器有里程计、IMU、激光、超声波、红外、相机、激光雷达等。激光、超声波、红外、激光雷达等属于主动式，其主动发射信号测量，使用多传感器时，容易相互干扰。里程计、IMU和相机等属于被动式多传感器时，使用多传感器时，不容易相互干扰。各部分硬件平台之间可以使用有线或无线通信。有线通信包括串口通信、SPI、I2C、HDMI等，无线通信包括TCP/UDP通信、Wifi、蓝牙、433等。

云大脑、PC和工控机一般用来执行图3-1应用组件中的算法，如SLAM、导航、计算机视觉（CV）、自然语言处理（NLP）、语音识别等相关算法。目前，CV、NLP、语音识别常用深度学习的方法，深度学习涉及庞大的矩阵运算，这就需要计算机带有具有并行运算能力的GPU。做深度学习一般用英伟达（NVIDIA）的GPU，其使用CUDA并行计算架构，方便使用TensorFlow、Pytorch等深度学习框架。当然，英伟达系列显卡越贵的运算速度越快，现在常用的一般是1080以上了。SLAM的优化中也可能涉及大的矩阵运算，也可以使用GPU加速。在深度学习任务中，GPU的运算速度是CPU的几十倍，甚至上百倍。云大脑具有服务器级别的计算能力，只要解决了通讯的问题，机器人就能方便、快速的从云大脑获取信息，可以将深度学习这种计算量庞大的运算交给云大脑，而5G的出现正好能有效的解决通信的问题。相信云大脑的加入能大大加速智能机器人的发展进程，是机器人具备真正的智能，比如在云大脑中存储生活中物品的特征及属性，机器人通过查询更容易识别和认知现实世界，从而，作出更好的决策，语义SLAM是一个很好的例子。

树莓派是一款卡片大小的处理器，具备计算机所有功能，计算能力接近笔记本电脑。如今，已经发行了树莓派4b，最高配具有4G运存，但不能安装Ubuntu 16.04兼容的Linux Min和Ubuntu MATE，使用ROS只能安装官方的Raspbian系统。之前的版本可以安装Ubuntu 16.04兼容的Linux Mint和Ubuntu MATE，基本操作和在Ubuntu 16.04上一样，可以方便地安装Ubuntu上的库。树莓派有usb接口，可以直接驱动带有usb接口的相机和激光雷达，SLAM和导航的算法也能在树莓派上处理，但运算速度亚于第一梯队的平台。另外，树莓派有接外设的引脚，可以实现单片机的大多数功能。Jetson Nano开发板和Jetson TX2开发板是英伟达发布的开发板，带有英伟达的显卡，体积也和微处理器差不多，可以安装在小型移动机器人上。这两款开发板也可以安装Ubuntu系统，易于安装和运行TensorFlow、Pytorch等深度学习框架下的深度学习算法，还能执行一些轻量级神经网络的训练任务。树莓派、Jetson Nano开发板和Jetson TX2开发板使机器人边缘计算能力大大提高，可以在机器人端执行大部分计算，它们既可以作上位机使用，也可以作下位机使用。

微处理器可以用于控制电机和采集传感器数据，一些简单的逻辑算法单片机也完全能够胜任了，如一款高空幕墙清洗机器人的主要算法可以使用STM32单片机完成，当然，不涉及图像处理部分。

控制电机常用PID控制，典型的是双闭环（速度环和电流环）控制。双闭环控制电路设计有2种方法，一种是硬件控制，通过调节电阻、电容电感等元器件阻抗大小调节PID值；另一种是软件控制，一般使用DSP，PID值调节方便，是目前主流的方法。

常用的里程计是编码器，编码器测量车轮的旋转量，并通过导航推测（dead reckoning）推算机器人的大致位置。在这种情况下会发生一定的误差，此时用IMU测得的惯性信息补偿位置信息的误差。里程计在短时间内能够提供较好估计，IMU对于短时间内的快速运动能够提供一些较好的估计。

激光的测量精度、范围和价格都高于超声波。超声波测距仪的测量精度是厘米级的，激光测距传感器的测量精度是毫米级的。超声波测距仪的测量范围通常在80米以内，而手持式激光测距传感器的测量范围最高可到200米，激光测距望远镜的测量范围更是多达几百几千米，甚至更远。激光测距精度与光强关系较大，只在一定光强范围测得距离是准确的，如激光在玻璃、反光很强或很弱的材质测得距离不准确，但超声波可以测得正确距离。

相机可分为单目相机、双目相机和深度相机。

日常生活中使用的一般是单目相机。其优点是结构简单，成本低，便于标定和识别。缺点是在单张图片里，无法确定一个物体的真实大小。它可能是一个很大但很远的物体，也可能是一个很近很小的物体。通过相机的运动形成视差，可以测量物体相对深度。但是单目SLAM估计的轨迹和地图将与真实的轨迹和地图相差一个因子，也就是尺度（scale），单凭图像无法确定这个真实尺度，所以称尺度不确定性。

双目相机代表性的产品包括Point Grey的Bumblebee相机和韩国InRobot公司的OjOcamStereo，以及小觅摄像头、Intel实感跟踪摄像头T265（适合室内）。其优点是基线距离越大，能够测量的距离就越远；并且可以运用到室内和室外。缺点是配置与标定较为复杂，深度量程和精度受到双目基线与分辨率限制，视差计算非常消耗计算资源，需要GPU/FPGA设备加速。

深度相机（Depth Camera）有多种名称，在类似LDS（laser distance sensor，激光距离传感器）的范畴内被称为Depth sensor，可以获得彩色图像时也被称为RGB-D camera，而微软公司成功普及的深度相机被称为Kinect Camera。根据获取信息的方法，Depth camera可以被分成多种类型，诸如ToF（Time of flight，飞行时间）、结构光（Structured Light）等。其优点是通过结构光或ToF的物理方法直接测量物体深度信息。缺点是测量范围窄，噪声大，视野小，易受日光干扰（红外对阳光敏感），无法测量透射材质等问题，主要用在室内，室外很难应用。

ToF方法是发送红外线后利用返回所需的时间测量距离。通常，IR发光部和收光部是成对的，并读取由每个像素测量的距离。ToF方法比后面将介绍的利用相干辐射模式的结构光方式更昂贵的原因是这种结构方面的原因提高了硬件的价格（最近，引入了使用相位差的距离计算方法，因此价格在下降）。采用ToF方式的传感器有Panasonic的D-IMAGER、MESA Imaging的SwissRanger、Fotonic的FOTONIC-B70、pmdtechnologies的CamCube和CamBoard、SoftKinectic的DepthSense DS系列以及微软最新发布的Kinect 2。

结构光方式的代表性产品是微软的Kinect和华硕的Xtion，它们使用相干辐射模式（pattern of coherent radiation，利用US20100225746专利）。此外，还有PrimeSense的Carmine和Capri以及最近的Occipital的Structure Sensor。这些传感器的共同点是都使用PrimeSense公司的PrimeSense片上系统（SoC）。使用PrimeSense公司的 PrimeSense SoC的Depth Camera是一款由一个红外投影仪和一个红外相机组成的传感器，它使用了现有的ToF方法中从未使用的相干辐射模式。该技术解决了现有ToF方式的硬件昂贵的问题和外部干扰等问题，因此备受关注。但是，苹果在2013年12月收购PrimeSense时出现了问题。PrimeSense的Carmine和Capri产品已经不再可用，而且微软的Kinect也停产，而华硕的Xtion也即将停产（库存除外）。Occipital公司的Structure Sensor是采用PrimeSense SoC的最后一款产品，目前是将此产品作为苹果的附件出售，但无法知道未来会发生什么。以低价流行的产品已经隐藏在历史中。

激光距离传感器（Laser Distance Sensor，LDS）有多种名称，比如激光雷达（LIDAR）、激光测距仪（Laser Range Finder，LRF）和激光扫描仪（Laser Scanner）。LDS是利用激光光源来测量与物体的距离的传感器。LDS传感器具有高性能、高速度和实时数据采集的优点，因此在距离测量方面有着广泛的应用。由于这些优点，它是在机器人领域被广泛使用的传感器，比如用于使用距离传感器的SLAM或用于识别人或物体识别。由于其优越的实时性能，最近还被广泛用于无人驾驶车辆。典型的产品是在室内广泛使用的Hokuyo的URG系列。多用于室外的产品有SICK和Velodyne的配有多个激光传感器的HDL系列。这些传感器最大的问题是价格。一般来说，不同产品的价格不尽相同，但大多是几千美元左右，而其中Velodyne的HDL系列是几十万元的产品。弥补这些缺点的中国产品（如RPLIDAR）以2800元左右的低价进入了市场，而近期则出现了韩国的一家公司推出的一款700多元的LDS（HLS-LFCD2）。国产的EAI产品更便宜，它的YDLIDAR X4只需499元。

1. **软件平台**

本节主要讨论SLAM、导航算法及相关的一些人工智能算法，其流程图如图3-3。当机器人走到未知环境时，它需要边定位边建图，需要在已有地图完成导航并探测未知区域。对于移动机器人来说，SLAM作为底层技术，服务于上层定位、导航、避障、重建和可视化等，稀疏地图可以用于机器人定位，但导航、避障和重建需要用到稠密地图。视觉SLAM可以用于建立稀疏、半稠密和稠密地图，一般越稠密计算量越大。



图3- 3 SLAM和导航流程图

SLAM按传感器可分为视觉SLAM和激光SLAM。视觉SLAM的优点是结构简单，安装的方式多元化，无传感器探测距离的限制，成本低，图像信息丰富，可提取语义信息；缺点是环境光影响大、暗处或无纹理区域无法工作，运算量大，构建地图本身难以直接用于路径规划和导航，传感器动态性能还需提高，地图构建时会存在累积误差。激光SLAM优点是可靠性高、技术成熟，建立的地图直观、精度高、不存在累积误差，地图可直接用于路径规划和导航；缺点是受雷达探测范围限制，安装有结构要求，地图缺乏语义信息。激光SLAM是目前比较成熟的定位导航方案，视觉SLAM是未来研究的主流方向。通常，在获得相机位姿后，可以使用点云库（PCL，Point Cloud Library）对激光雷达或深度摄像头获得的点云得到2D或3D的稠密地图。视觉和激光SLAM的融合也是不错的研究方向，视觉SLAM信息丰富，适合重定位，而激光SLAM建立的地图适合路径规划和导航。

SLAM可以按使用的方法可分为几何和学习两大类，图3-4。几何SLAM多使用文献[8]、[9]介绍的贝叶斯方法和多视图几何的方法。学习SLAM是几何SLAM和深度学习等人工智能方法的结合。



图3- 4 SLAM分类图

1. **视觉SLAM**

典型的SLAM系统如图3-5，传统的方法包括运动方程和观测方程，运动方程利用两帧图像计算相机位姿，观测方程用于后端优化。单目摄像头较难计算像素点的深度，需要通过两帧图像计算；双目可以通过单次拍摄计算深度；RGB-D摄像头可以直接获得深度。

* 视觉里程计：也称帧间估计，通过前后两帧图像计算摄像头位姿。
* 后端优化：后端接受不同时刻视觉里程计测量的相机位姿，以及回环检测信息，对它们进行优化，减小噪声的干扰和累积误差，得到全局一致性轨迹和地图。
* 回环检测：判断自身是否进入历史同一地点。回环检测发生时可触发 SLAM 后端全局一致性算法进行地图优化，消除累积轨迹误差和地图误差。闭环检测问题本质上是场景识别问题。
* 建图：根据估计的轨迹，建立与任务要求对应的地图。



图3- 5 视觉SLAM流程图

特征法首先通过特征提取和描述子匹配找出两帧图像相同的像素点，可以通过关键帧的位姿计算这些像素点的世界坐标，然后利用几何算法计算相机位姿。相机可以看成一个小孔成像的模型，单目相机具有尺度不确定性，需要通过两帧图像来计算像素点的深度。整个计算过程中是帧到帧累加计算，所以会有累积误差，这就需要利用后端和闭环检测来减小累积误差。后端采用图优化的方法，将一些路标点投影到相机成像画面，通过一些算法（如BA）使投影点和实际点尽可能接近。比较有代表性的特征有SIFT、SURF和ORB，目前ORB用的比较多，ORB-SLAM2兼容了单目、双目和深度相机，具有较好的实时性，目前应用比较广泛。得到位姿和深度后，建图还是比较容易，但单目使用这种方法只能建立稀疏的三维图，不能用于导航。

特征直接融合法在提取特征点或关键点后不使用描述子匹配，而是使用光流法来跟踪特征点的运动。

直接法也称光流法，有稠密和半稠密的方法。假设相邻两帧图像的像素值不变，通过梯度下降的方法调整位姿，使两幅图像对应像素点的方差最小，这就要用到李代数，因为旋转公式有约束条件，不好做梯度下降。可以进行稠密和半稠密建图。目前比较好的方法是Direct Sparse Odometry (DSO)。稠密地图可以用建立网格地图和八叉树地图，用于导航和避障。

滤波的方法将在激光SLAM中介绍，也可以用在视觉SLAM中。

人工设计的稀疏图像特征当前有很多局限性，一方面如何设计稀疏图像特征最优地表示图像信息依然是计算机视觉领域未解决的重要问题，另一方面稀疏图像特征在应对光照变化、动态目标运动、摄像机参数改变以及缺少纹理或纹理单一的环境等方面依然有较多挑战。

几何方法在定位和建图具有比较好的准确性和实时性，但鲁棒性不太好，会面临特征稀少、光强变化、尺度不确定性、相机内参漂移等问题，将深度学习应用到SLAM可提高其鲁棒性、语义信息以及学习能力，可以使机器人更好地与环境交互。CNN可以提取大量的隐性图像特征，具有尺度不变性、旋转不变性。CNN、RNN、LSTM、encoder-decoder、无监督学习在视觉SLAM方面都有应用，引入了CV和NLP方面的大多数模型。文中主要提到深度学习在SLAM的深度估计、位姿估计、语义SLAM（如图3-6）三个方面结合比较多。



图3- 6 语义SLAM图

目前深度学习结合SLAM有3个方向：

* 用深度学习方法替换传统SLAM一个或几个模块特征提取，如特征匹配，提高特征点稳定性，提取点线面等不同层级的特征点，深度估计，位姿估计，重定位；
* 在传统SLAM之上加入语义信息，如图像语义分割，语义地图构建；
* 端到端的SLAM，端到端是输入image输出action，没有定位和建图。 如机器人自主导航（深度强化学习）等。

监督学习需要数据集来驱动，但不同机器人面临不一样的未知环境，很难建立通用的数据集，建立数据集也需要花费大量的人力和时间，这是当前面临的瓶颈。无监督学习、传统方法与深度学习的结合、多传感器（雷达、超声波、激光、陀螺仪）的融合有利于解决目前面临的一些问题。常用的数据集如下。

* 深度估计：KITTI, TUM, NYU。
* 重定位：7-scenes dataset，Cambridge landmarks。
* 视觉里程计（VO）：KITTI, Robotcar,M’alaga, EuRoc MAV, NYU,TUM。
* 场景分割：PASCAL VOC,NYU,Synthia,Cityscapes, KITTI,ADK20。

接下来简单介绍一下ORB-SLAM2,它是经典的特征法，涉及了几何SLAM中大多数方法。其涉及的主要文献是文献[16]-[31]。文献[16]-[18]是ORB-SLAM2改进过程中产生的三篇论文；文献[19]-[22]是描述ORB特征的；文献[23]介绍词袋方法，可以通过图像快速搜索其中的特征，也可以通过特征快速找到出现过该特征的图像，在重定位和闭环检测时能快速识别以前经过的场景；文献[24]介绍图优化方法，可以在跟踪时求解位姿，以及优化局部和全局地图；文献[25]介绍重定位和闭环检测的方法，重定位是3D-2D的问题，闭环检测是3D-3D的问题；文献[26]用于初始化后的跟踪和重定位中3D-2D的问题；文献[27]用于求解闭环检测中的3D-3D的问题；文献[28]在跟踪线程中局部地图优化用到；文献[29]、[30]用于闭环检测中计算相似变换和位姿图优化；文献[31]用于从单应矩阵H中分解出位姿；文献[9]中有从本质矩阵E中分解出位姿的方法以及随机抽样一致 (RANSAC，Random Sample Consensus)算法。

ORB-SLAM2的整体框图如图3-7。为了实现实时性，ORB-SLAM将整个程序分为tracking、local mapping、loop closing3个线程。各线程的作用如下。

* Tracking：根据摄像头传来的帧进行实时定位和决定何时插入关键帧。关键帧是摄像头所有帧中比较重要的一些帧。
* Local mapping：处理新的关键帧及优化局部地图来减小累积误差。
* Loop closing：检测回环，进行full BA减小累积误差，建立全局一致地图。

ORB-SLAM2具有有以下几个优点。

* 整个系统使用同一种特征：ORB特征，使其更加高效、简单和可靠；
* 可以在大的环境中实时运行；
* 在essential graph上实现实时闭环检测，优化位姿；
* 实时重定位，具有很好的视角和光照不变性；
* 健壮的自动初始化；
* 剔除冗余的帧和地图点，这也是使其能在环境中长期运行的原因。



图3- 7 ORB-SLAM2整体框图

常用的开源SLAM方案如表3-1所示。除了这些开源方案之外，还能在openslam.org之类的网站上找到许多其他的工作，例如DVO-SLAM，RGBD-SLAM-V2，DSO以及一些 Kinect Fusion 相关的工作等等。

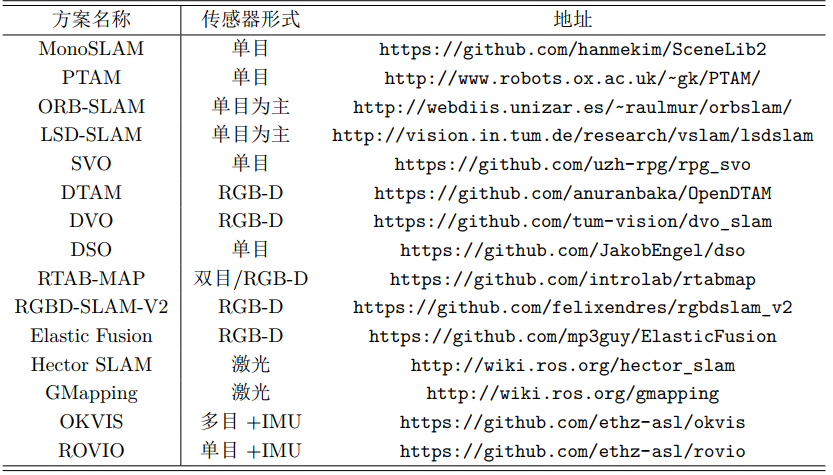


表3- 1 常用开源SLAM方案

看过了一些现有的方案，我们再来讨论一些未来的发展方向。大体来说，SLAM 将来的发展趋势一共有两个大类：一是往轻量级、小型化方向发展，让 SLAM 能够在嵌入式或手机等小型设备上良好的运行，然后考虑以它为底层功能的应用。毕竟大部分场合中，我们的真正目的都是实现机器人、AR/VR 设备的功能，比如说运动、导航、教学、娱乐，而 SLAM 是为上层应用提供自身的一个位姿估计。在这些应用中，我们不希望 SLAM 占据所有计算资源，所以对 SLAM 的小型化和轻量化有非常强烈的要求。另一个方面，则是利用高性能计算设备，实现精密的三维重建、场景理解等功能。在这些应用中，我们的目的是完美地重建场景，而对于计算资源和设备的便携性则没有多大限制。由于可以利用GPU，这个方向和深度学习亦有结合点。SLAM 的未来有视觉和IMU融合、语义SLAM两个大方向，还有基于线/面特征的 SLAM、动态场景下的SLAM、多机器人的等等。

1. **激光SLAM**

激光SLAM利用激光雷达采集的点云数据边定位边建图。下列是一些常用的激光SLAM算法。

* Hector SLAM(1900-2000年)：是基于EKF(Extended Kalman Filter,扩展卡尔曼滤波器)的方法，不依赖里程计，只根据激光信息便可建图。帧间匹配也不需要里程计，纯靠优化的方法，使用梯度优化方式。可以很好的在空中机器人、手持构图设备及特种机器人中使用，基于深度数据估计里程计信息，建图的稳定性没那么好。在特征不太明显的地方容易出现长廊效应。
* Fast SLAM（2002-2003年）：是基于RBPF（Rao-Blackwellised Particle Filter）的方法，同时集成了EKF方法。帧间匹配使用了3种优化方法：提升建议分布采样的质量；减少重采样次数，用一个量来表示当前估计和目标分布的差异性；考虑最近一帧的观测，把建议分布限制在一个狭小的有效区域，然后正常的对建议分布进行采样。
* Gmapping（2007年）：是基于RBPF的方法，依赖里程计，需要深度信息、IMU信息，打滑会造成里程计数据有错，从而使建图效果不好。帧间匹配使用Fast SLAM帧间匹配的前两种方法，加入了scan-match，是一种近似梯度下降的方法。
* Optical RBPF（2010年）:是一种基于RBPF的方法。是MRPT（Mobile Robot Programming Toolkit，移动机器人编程工具箱）上的一个开源算法，改进了Gmapping的方法。
* Karto SLAM（2010年）：是基于图优化的方法。帧间匹配使用相关匹配方法和分枝定界加速。运行时容易崩溃。
* Cartographer（2016年）：是谷歌发布的基于图优化的方法，带有回环检测，适用2D/3D建图。帧间匹配使用CSM（粗搜索）和梯度优化（细搜索）结合的方法。目的是在计算资源有限的情况下，实时获取相对较高精度的2D地图。考虑到基于模拟策略的粒子滤波方法在较大环境下对内存和计算资源需求较高，其采用基于图网络的优化方法。

下面简单介绍一下Gmapping和Cartographer算法，它们都有现成的ROS功能包。

Gmapping算法涉及的主要文献是文献[33]-[43]和文献[8]。文献[33]、[34]是Gmapping改进过程产生的两篇论文；文献[35]、[36]描述RBPF方法；文献[37]、[38]分别描述Gmapping参考的算法Fast SLAM 1.0和Fast SLAM 2.0；文献[39]、[40]描述Gmapping中使用的扫描匹配方法；文献[41]描述栅格地图方法；文献[42]描述了改进建议分布的方法；文献[43]使用里程计数据做出位姿的初始估计；文献[8]的第5章和第6章分别描述了Gmapping使用的运动模型和感知模型。

Gmapping是一种改进的基于栅格地图的slam算法，也是对Fast SLAM的改进，它的建议分布与Fast SLAM 2.0相似。使用了具有自适应自适应建议分布和选择性重采样的RBPF，每个粒子携带一个单独的环境地图。该方法可以减少获取栅格地图的粒子数量。常见RBPF步骤如下：

1. 采样：根据建议分布从上一时刻粒子集采样生成新的粒子集；
2. 计算重要性权重：计算每个粒子的重要性权重，其值为目标发布与建议分布的商；
3. 重采样：根据重要性权重对粒子集进行采样，权重大的粒子更可能保留下来。采样完后，使所有粒子的权重相等；
4. 地图估计：根据上面求得的轨迹和所有观测计算地图。

RBPF存在2个缺点：

1. 复杂性，这是根据构建精确地图所需的粒子数量来衡量的。减少粒子数是主要的挑战。
2. 重采样过程可能会消除正确的粒子。

Gmapping提出解决上述问题的方法如下。

1. 提出更准确的建议分布，将运动和测量都考虑了，提高了粒子滤波预测步骤的准确性，通过评估由扫描注册程序获得的与粒子相关的最可能位姿的可能性来计算的。在生成新粒子时，最后的读数将被考虑在内，从而可以根据一个更可靠(因此也更准确)的模型来估计系统的演化，这个模型比仅使用Fast SLAM 1.0中最后的里程计读数得到的模型更准确；
2. 自适应重采样策略，只允许在需要时执行重采样步骤，从而保持合理的粒子多样性。监测Neff控制的自适应重采样，可以减少粒子消失的风险。Neff越小，权重差别越大。

Gmapping的具体步骤如下。

1. 根据文献[43]的方法使用里程计数据做出位姿的初始估计；
2. 根据上一步位姿估计和上一帧扫描地图进行扫描匹配，得到最佳位姿估计。若扫描匹配失败，则使用运动模型估计位姿，直接跳到第5）步；
3. 在上一步最佳估计附近采样K个粒子的位姿，每个粒子的均值和方差由测量和运动联合的模型计算，其中归一化因子根据文献[44]求出；
4. 根据上一步计算的每个粒子的均值和方差的高斯分布采样K个粒子的位姿，作为新的粒子集；
5. 使用递归的方法更新权重，当Neff小于阈值时进行重采样；
6. 根据测量和位姿估计地图。

Cartographer主要参考文献[45], Cartographer 能产生一个精度为5cm的2D栅格地图。其在前端匹配环节区别与其它建图算法的主要是使用了Submap这一概念，每当获得一次laser scan的数据后，便与当前最近建立的Submap去进行匹配，使这一帧的laser scan数据插入到Submap上最优的位置（这里用的是高斯牛顿解最小二乘问题）。在不断插入新数据帧的同时该Submap也得到了更新。一定量的数据组合成为一个Submap，当不再有新的scan插入到Submap时，就认为这个submap已经创建完成，接着会去创建下一个submap。因此这里scan matching的本质是当前laser scan与多个邻近的laser scan之间进行匹配。

通过scan matching得到的位姿估计在短时间内是可靠的，但是长时间会有累积误差。因此Cartographer应用了回环检测对累积误差进行优化。所有创建完成的submap以及当前的laser scan都会用作回环检测的scan matching。如果当前的scan和所有已创建完成的submap在距离上足够近，则进行回环检测。这里为了减少计算量，提高实时回环检测的效率，Cartographer应用了branch and bound(分支定界)优化方法进行优化搜索。如果得到一个足够好的匹配，则会将该匹配的闭环约束加入到所有Submap的姿态优化上。

激光SLAM已经比较成熟，学习激光SLAM主要为了建立环境的栅格地图用于导航，以及熟悉激光SLAM的使用，便于从事视觉和激光SLAM融合算法的研究。

1. **导航**

机器人导航就是如何引导机器人朝着某一目标前进的问题。导航可以分为反应式导航和基于地图的类人导航方法。反应式导航在完成任务时不依赖任何地图，例如机器人寻着光源前行、沿地上白线行走、沿墙壁穿越迷宫以及沿随机路径完成房间吸尘等。更先进的机器人则使用基于地图的类人导航方法，这种方法可以支持更复杂的机器人任务。接下来主要讨论第二种方法。机器人实现自主导航至少需要4个条件：地图；测量或估计机器人的姿态的功能；识别障碍物，如墙壁和物体的功能；能够计算出最优路线并行驶的功能。

以ROS中的导航功能包为例，图3-8是有关导航堆栈配置的各必要节点和话题的关系图。导航元功能包主要包括以下三个功能包。

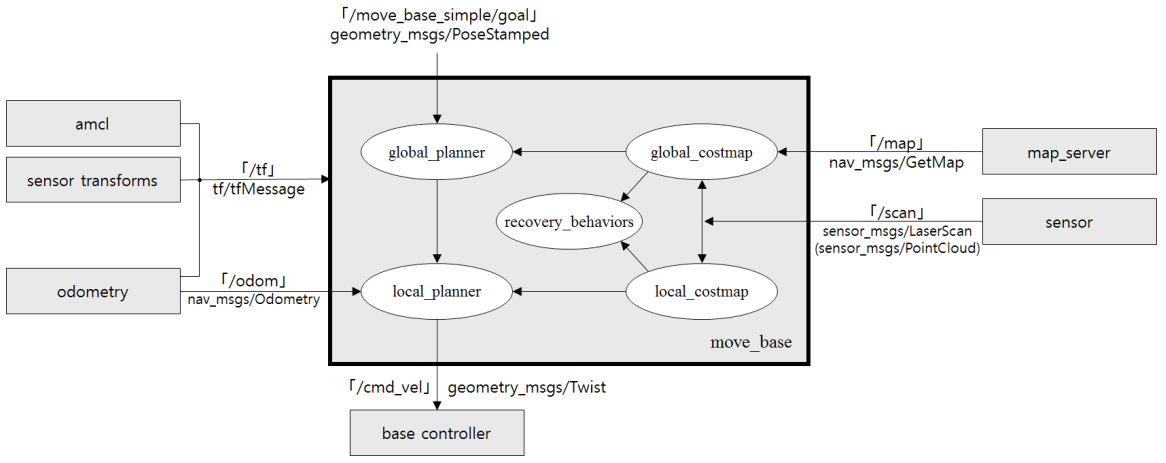


图3- 8 有关导航堆栈配置的各必要节点和话题的关系图

* map\_server：用于加载先前创建的地图。
* amcl（自适应蒙特卡罗定位，Adaptive Monte Carlo Localization）：实现二维地图中的机器人定位。
* move\_base：实现机器人导航中的最优路径规划。包括全局路径规划（global\_planner）和本地实时规划（local\_planner）,前者使用Dijkstra或A\*算法进行全局路径规划。在实际情况中，机器人往往无法严格按照全局路线行驶，所以需要针对地图信息和机器人附近随时可能出现的障碍物规划机器人每个周期内应该行驶的路线，使之尽量符合全局最优路径，后者使用DWA(Dynamic Window Approaches)算法搜索躲避和行进的多条路径，综合各平均标准（是否会撞击障碍物，所需时间等）选取最优路径，并且计算行驶周期内的线速度和角速度，避免与动态出现的障碍物发生碰撞。DWA算法包含了图3-3中的局部规划和控制。

全局路径规划给出通往目的地的路径，主要包括如下算法。

* Dijkstra：也称单源最短路径算法。算法的基本思想是：每次找到离起点最近的一个顶点，然后以该顶点为中心进行扩展，最终得到起点到其余所有点的最短路径。主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。
* A\*：是Dijkstra的改进版。与Dijkstra区别在启发函数，待处理队列排队时在已走过的距离上加入对未来预期要走的距离作为排队依据。Dijkstra均匀往外搜索，A\*会根据未来预期距离有目的的往外搜索。
* D\*：是A\*的升级版。允许成本地图改变后重新计算路径，但它不支持改变目标。若B为目标，那以B为中心产生规划，然后根据不同起点A进行查询。
* 概率路线图方法（PRM,Probabilistic road map）:在无障碍物空间中随机采样n点；把n个点按照可连通性连接，获得概率路线图G；设定起点和终点，分别在G中找出离起点和终点最近的图节点V；以这两个节点为起点和终点，然后使用A\*或Dijkstra算法搜索路径。其优点是最简单；缺点是线路非最短，狭长处较难采样。
* 沃罗诺伊（Voronoi）：取出地图中空白区域；细化空白区域获得骨架图G；设定起点和终点，分别在G中找出离起点和终点最近的骨架点V；以两个骨架点为起点和终点，然后使用A\*或Dijkstra算法搜索路径。优点是完整的地图骨架网络，最安全的线路；缺点是线路非最短。
* 可视图（visibility graph）：取出图中空白区域；将空白区域边界角点V取出，两两间按连通性连成G；把起点和终点作为节点按连通性加入G；使用A\*或Dijkstra算法搜索路径。其优点是线路最短；缺点是不安全。
* 快速扩展随机树（RRT，rapidly exploring tree）：以起点为根节点初始化树T；随机采样位姿，按可达条件加入到树T的叶子；重复上一步直到目的地加入，为树T的叶子。其优点是直接考虑了控制；缺点是路径非最短，慢，路径会很奇怪，狭长处较难采样。
* 蚁群算法：用蚂蚁的行走路径表示待优化问题的可行解，整个蚂蚁群体的所有路径构成待优化问题的解空间。路径较短的蚂蚁释放的信息素量较多，随着时间的推进，较短的路径上累积的信息素浓度逐渐增高，选择该路径的蚂蚁个数也愈来愈多。最终，整个蚂蚁会在正反馈的作用下集中到最佳的路径上，此时对应的便是待优化问题的最优解。

局部路径规划是在不走错路的前提下，躲避行走中遇到的障碍物，主要包括如下算法。

* 人工势场法（APF， artificial potential field）：设计目的地和障碍物的势场公式；用势场梯度计算机器人的参考速度。其优点是不需要再计算轨迹，计算量小；缺点是易陷入局部最低点，无法保证绝对无碰撞，狭窄处无法通过，狭窄通道中摆动。
* 向量场直方图（VFH，vector field histogram）：根据不同角度方向障碍物距离计算直方图；根据直方图，阈值以下的空间即为安全空间。其优点是计算量小；缺点是阈值选择对避障影响大（半径），没考虑运动限制。
* 动态窗口法（DWA，dynamic window approach）：根据当前速度、机器人加速度和时间步（将一条轨迹分成离散的很多步）限制划出采样空间，大量采样不同轨迹，抛弃有碰撞的轨迹；对无碰撞的轨迹进行评分，选取获得分数最高的轨迹。

其优点是考虑了轨迹和碰撞；缺点是计算量较大，调参困难，容易忽略狭窄通道。

在一些局部路径规划算法后，还需要根据其产生的路径生成适合机器人行走的控制信号，这就需要考虑运动学和动力学方面知识。

以上提到的算法已经比较成熟了，强化学习在导航中的应用还可以进一步研究。

1. **相关人工智能算法**

我认为智能机器人的终极版会融合大多数人工智能算法。以下是自己调研过的一些方向，当然，还有很多其他方向可以与机器人结合。

* 多智能体：多智能体系统由一群有自主性的，可互相交互的实体组成，它们共享一个相同的环境，通过感知器感知环境并通过执行器采取行动。多智能体在现实生活中已有应用，如机器人战队，分布式控制和资源管理。在多智能体系统中会面临利益冲突、信息有限和回报等问题，多智能体环境可分为合作环境和竞争环境。在合作环境下，智能体注重集体利益和社会利益，在竞争环境下智能体注重个体利益。在多智能体系统中需要均衡Fairness, Social Optimality和Individual Rationality。多智能体主要应用的技术是强化学习（RL，Reinforcement Learning）和深度强化学习（DRL, Deep Reinforcement Learning）。DRL与NLP、机器人、多智能体等很多技术的结合展现出很好的发展前景。RL与DL（DL，Deep Learning）等技术的融合能有效增强算法的学习能力。
* EQA(Embodied Question Answering)：机器人在未知环境中以第一视角来回答问题，机器人收到问题后，通过在房间寻找对应的目标然后回答问题。这个任务非常新，结合了CV、NLP、DL，难度很高，准确率很低。图3-9是文献[50]的模型，问题是“浴缸的颜色和浴室的水池一样吗？”,那么机器人必须找到浴缸和水池，然后判断他们的颜色是否一样。由图3-9可以发现program generator和VQA module作为输入、输出模块相对独立，program generator将question分解为几个子程序，VQA module则用于比较controller两个输出属性产生answer。controller比较当前第一视角图像（经CNN处理后）与子程序中所要求图像进行比对，判断是否到达目标房间或找到目标，逐步完成子程序所要求的任务，并给navigator模块提供当前子程序的目标。Navigator模块则根据controller输出的target、当前第一视角图像（经CNN处理后）以及上一个动作产生相应的动作，寻找controller给定target。过程中，寻找房间和物体的过程是不一样的，房间是通过12张全景图来分辨的，而物体是通过语义信息来分辨的，CNN是预训练好的。另外，navigator的动作为左转30度、右转30度和前进。作者发现较EQA-v1转角的增大，减少的动作的次数，使得agent更容易训练。对于整个模型的训练过程，作者没有细说，只是简单提到可以使用了模仿学习（Imitation Learning IL)或强化学习（RL）。

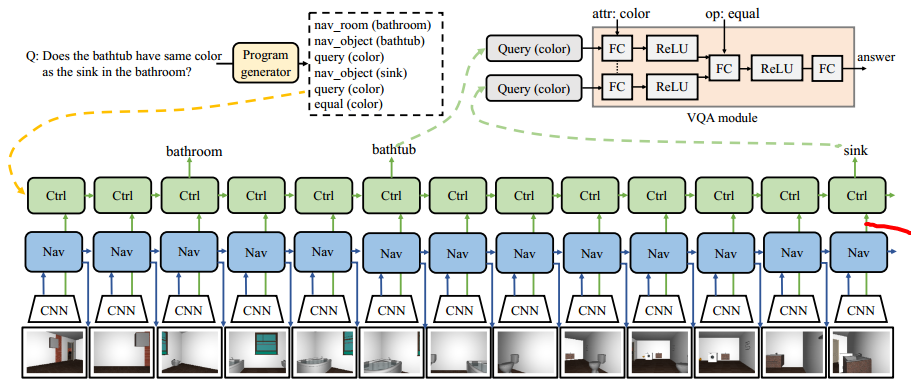


图3- 9 EQA模型图

* VLN（Viusal-and-Language Navigation）:VLN和VQA的区别是它们输入都是图片和句子，但VQA输出是句子，VLN输出是一系列动作。具体方法可以参考文献[51]。
* 图像问答（VQA, Visual Question Answering）：VQA有机结合了计算机视觉和自然语言处理（NLP）。它与其它计算机视觉任务最大的区别是问题没有预先设定，是在运行时设定的。图像问答与文本问答有很大相似之处，文本问答的很多方法也能用在图像问答上。图像问答相较于文本问答，将输入的文本换成了图像，这就带来了很多挑战，比如维度变高、噪声变多、图像中包含现实生活的很多信息。相较于图像描述，图像问答可能涉及不包含在图像中的常识。VQA的4个主要方法：联合嵌入方法、注意力机制、组合模型、使用外部知识库的模型。大多数VQA模型都要用到联合嵌入方法，一个简单模型如图5-10所示。

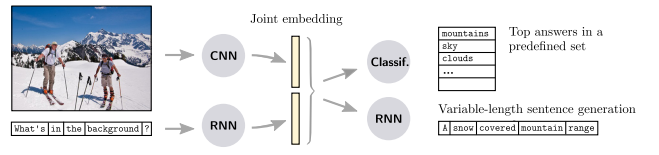


图3- 10 VQA模型图

* 对话：机器人像人一样智能，还必须必备对话的功能。人与人之间可以通过文字和语音交流。使用文字交流涉及NLP相关知识。使用语音交流涉及语音识别相关知识，也可以使用pipeline的形式，先将语音转换成文字，再用NLP技术来处理。NLP和语音识别有些相似之处，语音和文字只是用于表达的不同形式，它们都是串行的，和可以并行处理的图像数据不同。NLP中的对话系统可分任务型和非任务型。任务型有pipeline（如图3-11）和端到端的方法，非任务型有生成式、检索式和混合式。各大互联网公司和手机厂商都推出了自己的聊天机器人，比如微软小冰、阿里的天猫精灵、小米的小爱等。如果给他们加上可以行走的身体，并能通过对话让他们完成一些任务，那就具备了人的基本智能。

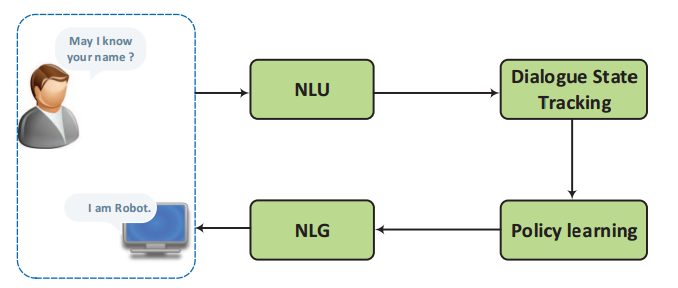


图3- 11 任务型对话系统

1. **应用方案**

结合实验室已有的设备和调研结果，最后搭建了如图4-1所示的机器人（激光雷达的安装架子待完善），机器人的电气原理图如图4-2。其中，树莓派用的树莓派4b，摄像头用的是微软的kinect 2.0，激光雷达用的是EAI的YDLIDAR X4，CAN分析仪和直流电机驱动器都是国产品牌。



图4- 1 两轮自主机器人实物图



图4- 2 两轮自主机器人电气原理图

软硬件的设计主要参考TurtleBot3。为了避免摄像头和激光雷达数据从树莓派发送到笔记本电脑的延迟，直接将摄像头和激光雷达装在笔记本电脑上。另外，目前只使用了摄像头、激光雷达、里程计、IMU和红外接收头等传感器，还未安装激光和超声波，使用树莓派作为下位机控制电机和采集里程计、IMU数据、红外接收头等传感器数据已经足够了。如果后期需要安装较多激光、超声波等传感器，可以在树莓派后增加微处理器来读取传感器数据，树莓派很方便增加外设或与其他微处理器通信。

笔记本电脑和树莓派分别安装Ubuntu 16.04和Raspbian系统。同时，两者都安装了ROS系统，将树莓派和笔记本电脑接入同一个局域网，只需简单配置ROS环境就能实现树莓派与笔记本电脑之间通信，也很方便调用ROS已有的功能包。

最后，通过调试，完成了ROS系统的搭建、对电机的控制、各传感器数据的采集以及树莓派与笔记本之间的通信，实现了遥控功能。接下来主要工作是通过该平台验证一些已有的视觉SLAM算法，并加以研究和创新。

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)[M].电子工业出版社.2013.09.
5. 伊恩 古德费洛.深度学习[M].人民邮电出版社.2017.08
6. 郑泽宇.TensorFlow实战Google深度学习框架(第2版)[M].电子工业出社.2018.01.
7. Magnus Lie Hetland.Python基础教程（第3版）. 人民邮电出版社.2018.02.
8. 塞巴斯蒂安 特龙.概率机器人[M].机械工业出版社.2019.09.
9. 韦穗.计算机视觉中的多视图几何[M].安徽大学出版社.2002.08
10. 高翔.视觉SLAM十四讲（从理论到实践）[M].电子工业出版社.2017.03.
11. 赵洋,刘国良,田国会, et al.基于深度学习的视觉SLAM综述[J].机器人,2017(6).
12. Ruihao L,Sen W,Dongbing G.Ongoing Evolution of Visual SLAM from Geometry to Deep Learning:Challenges and Opportunities[J].Cognitive Computation, 2018.
13. Real-Time Semantic Mapping of Visual SLAM Based on DCNN.
14. Tateno K , Tombari F , Laina I , et al. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction[J]. 2017:6565-6574.
15. Cadena C , Carlone L , Carrillo H , et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age[J]. IEEE Transactions on Robotics, 2016, 32(6):1309-1332.
16. Mur-Artal R , Juan D. Tardós. ORB-SLAM: Tracking and Mapping Recognizable Features[C]// Workshop on Multi VIew Geometry in RObotics (MVIGRO) - RSS 2014. 2014.
17. Mur-Artal R , Montiel J M M , Tardos J D . ORB-SLAM: a Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.
18. Mur-Artal R , Tardos J D . ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2017:1-8.
19. Rosten E , Drummond T . Machine Learning for High-Speed Corner Detection[J]. 2006.
20. Rosten E , Porter R , Drummond T . Faster and better: a machine learning approach to corner detection[J]. IEEE Trans Pattern Anal Mach Intell, 2008, 32(1):105-119.
21. Calonder M , Lepetit V , Strecha C , et al. BRIEF: Binary Robust Independent Elementary Features[C]// Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV. 2010.
22. Rublee E , Rabaud V , Konolige K , et al. ORB: An efficient alternative to SIFT or SURF[C]// 2011 International Conference on Computer Vision. IEEE, 2012.
23. Galvez-LoPez D , Tardos J D . Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28(5):1188-1197.
24. g2o: A General Framework for Graph Optimization.
25. Raúl Mur-Artal, Juan D Tardós. Fast Relocalisation and Loop Closing in Keyframe-Based SLAM[C]// IEEE International Conference on Robotics & Automation. IEEE, 2014:846-853.
26. Moreno-Noguer F , Lepetit V , Fua P . Accurate Non-Iterative O(n) Solution to the PnP Problem[J]. Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision, 2007.
27. Arun K S . Least-squares fitting of two 3-D point sets[J]. IEEE Trans. Pattern Anal. Machine Intell. 1987, 9.
28. Strasdat H , Davison A J , Montiel J M M , et al. Double window optimisation for constant time visual SLAM[C]// IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011. IEEE, 2011.
29. Scale Drift-Aware Large Scale Monocular SLAM.
30. Closed-form solution of absolute orientation using unit quaternions.
31. Faugeras O D F L , Lustman F . Motion and Structure from Motion in a Piecewise Planar Environment[J]. International Journal of Pattern Recognition and Artificial Intelligence, 1988, 02(3).
32. Anthony Williams.C++并发编程实战[M].
33. Grisetti G , Stachniss C , Burgard W . Improving Grid-based SLAM with Rao-Blackwellized Particle Filters By Adaptive Proposals and Selective Resampling[C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18-22, 2005, Barcelona, Spain. IEEE, 2005.
34. Grisetti G , Stachniss C , Burgard W . Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23(1):34-46.
35. A. Doucet, J. de Freitas, K. Murphy, and S. Russel, “Rao-Blackwellized particle filtering for dynamic Bayesian networks,” in Proc.Conf. Uncertainty Artif. Intell., Stanford, CA, 2000, pp. 176–183.
36. K. Murphy, “Bayesian map learning in dynamic environments,”in Proc. Conf. Neural Inf. Process. Syst., Denver, CO, 1999, pp.1015–1021.
37. D. Hähnel, W. Burgard, D. Fox, and S. Thrun, “An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” in Proc. IEEE/RSJ Int. Conf. Intell.Robots Syst., Las Vegas, NV, 2003, pp. 206–211.
38. M. Montemerlo, S. T. D. Koller, and B. Wegbreit, “FastSLAM 2.0:An improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” in Proc. Int. Conf. Artif. Intell.,

Acapulco, Mexico, 2003, pp. 1151–1156.

1. M. Montemerlo, N. Roy, S. Thrun, D. Hähnel, C. Stachniss, and J.

Glover, CARMEN—The Carnegie Mellon Robot Navigation Toolkit 2002 [Online]. Available: <http://carmen.sourceforge.net>

1. N. Roy, M. Montemerlo, and S. Thrun, “Perspectives on standardization in mobile robot programming,” in Proc. IEEE/RSJ Int. Conf. Intell.Robots Syst., Las Vegas, NV, 2003, pp. 2436–2441.
2. H. Moravec, “Sensor fusion in certainty grids for mobile robots,” AIMag., pp. 61–74, 1988.
3. A. Doucet, N. de Freitas, and N. Gordan, Eds., Sequential Monte-Carlo

Methods in Practice. New York: Springer-Verlag, 2001.

1. F. Lu and E. Milios, “Globally consistent range scan alignment for

environment mapping,” J. Auton. Robots, vol. 4, pp. 333–349, 1997.

1. Doucet, “On sequential simulation-based methods for Bayesian filtering,” Univ. Cambridge, Dept. Eng., Signal Process. Group, Cambridge, U.K., Tech. Rep., 1998.
2. Hess W , Kohler D , Rapp H , et al. Real-Time Loop Closure in 2D LIDAR SLAM[C]// 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016.
3. Peter Corke.机器人学、机器视觉与控制—MTALAB算法基础[M].电子工业出版社.2016.05.
4. 周志华.机器学习[M].清华大学出版社.2017.08.
5. K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," in IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26-38, Nov. 2017.
6. Hao. INTERACTIONS IN MULTIAGENT SYSTEMS: FAIRNESS, SOCIAL OPTIMALITY AND INDIVIDUAL RATIONALITY[M]// Interactions in Multiagent Systems: Fairness, Social Optimality and Individual Rationality. Springer Publishing Company, Incorporated, 2016.
7. Yu L , Chen X , Gkioxari G , et al. Multi-Target Embodied Question Answering[J]. 2019.
8. Anderson P , Wu Q , Teney D , et al. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments[J]. 2017.
9. Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, Anton van den Hengel,Visual question answering: A survey of methods and datasets,Computer Vision and Image Understanding,Volume 163,2017,Pages 21-40,ISSN 1077-3142.
10. Chen H , Liu X , Yin D , et al. A Survey on Dialogue Systems[J]. ACM SIGKDD Explorations Newsletter, 2017, 19(2):25-35.
11. Li J , Galley M , Brockett C , et al. A Persona-Based Neural Conversation Model[J]. 2016.

