



# 机器人ROS开发实践

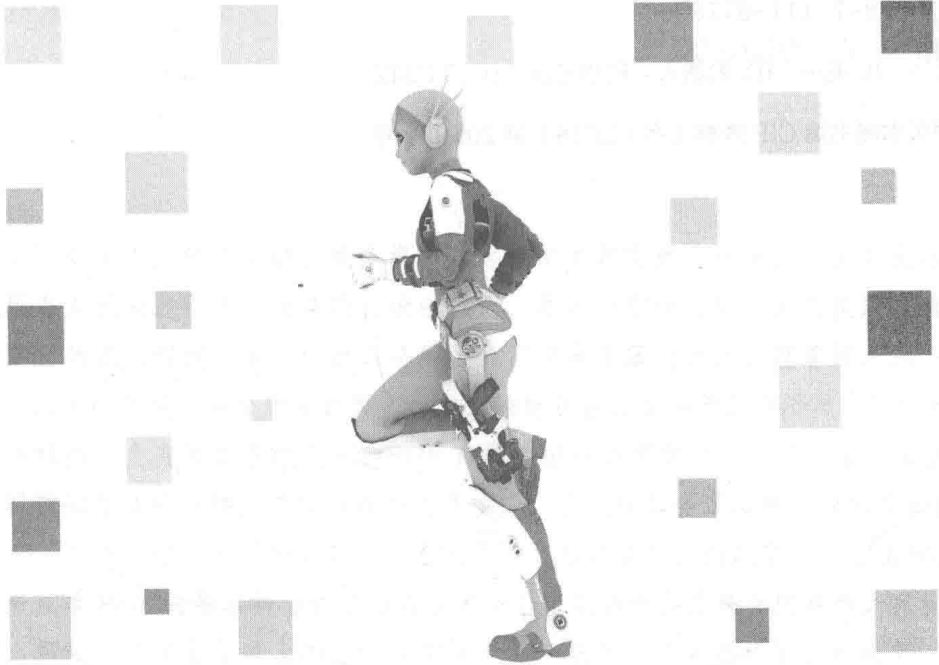
Robotics Programming Practice with ROS

无为斋主 编著



机械工业出版社  
China Machine Press

非外借



# 机器人ROS开发实践

无为斋主 编著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

机器人 ROS 开发实践 / 无为斋主编著. —北京: 机械工业出版社, 2019.1  
(机器人设计与制作系列)

ISBN 978-7-111-61720-4

I. 机… II. 无… III. 机器人—程序设计 IV. TP242

中国版本图书馆 CIP 数据核字 (2018) 第 296922 号

## 机器人 ROS 开发实践

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张志铭

印 刷: 三河市宏图印务有限公司

开 本: 186mm×240mm 1/16

书 号: ISBN 978-7-111-61720-4

责任校对: 殷 虹

版 次: 2019 年 1 月第 1 次印刷

印 张: 14

定 价: 69.00 元



凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

华章科技  
HZBOOKS | Science & Technology



# Preface | 前言

随着关键器件的研究突破，服务机器人这几年发展得如火如荼，无论是担当物品运输的物流机器人还是进行室内清洁的扫地机器人，是执行空中侦察的无人机还是承载人类自动驾驶梦想的无人驾驶汽车，机器人总能为我们带来无限的遐想。随着技术的不断成熟和商业推广的不断开展，一个曾被认为是科幻中的世界也许某一天就会展现在我们面前。

作为机器人主流操作系统之一，ROS 目前已经成为事实上的行业标准，同 Android 在智能手机领域的地位一样。但作为承载人类能听、会说以及会运动等特性的设备助理大脑，ROS 在涉及较多跨学科知识的同时，其系统化的文档却较少，因此学习门槛相对较高。

本书注重 ROS 的基本概念和核心内容，努力为读者提供系统化的视角，同时从开发实践出发，希望能让读者快速上手，迅速掌握开发一款能听会说的机器人的技能。由于机器人导航相关的导航算法等内容已经逐渐由专业的底盘厂商提供，同时导航涉及较多的专业算法和跨学科知识，考虑到本书的篇幅、专业性以及阅读对象，导航相关的内容在本书中不做过多介绍，有兴趣的读者可以查阅相关的资料和书籍进行学习。

与市面上 ROS 相关书籍（主要介绍较旧的 ROS Indigo、ROS Kinetic）不同，本书中的相关代码和环境搭建依托于当前最新的 Ubuntu 18.04 和 ROS Melodic 而进行，力图为读者提供最新的专业知识和行业解读。

## 读者对象

本书主要适合于有一定 C++ 基础、对机器人开发感兴趣但尚无过多接触的软件工程师和理工科学生。

## 如何阅读本书

本书共包含 9 章和 1 个附录，具体如下。

第 1 章介绍了机器人的发展历史以及 ROS 的发展历程，并介绍了如何搭建 ROS 环境。

第2章从一个简单的实例开始介绍了 ROS 的基本概念，如节点、话题、消息、服务、动作等。

第3章从一个基本的程序开始介绍了如何开发 ROS 程序包，以及如何编译、调试 ROS 程序包。

第4章介绍了如何进行 ROS 的核心技能（消息、服务、动作、启动等）开发，以及如何进行参数配置和分布式计算。

第5章介绍了 ROS 和运动控制相关的坐标系、控制器等，为读者学习定位、导航等打下基础。

第6章介绍了 ROS 如何实现与语音交互相关的语音识别、语义合成、声源定位等，这些内容有助于读者开发会说的机器人。

第7章介绍了 ROS 如何与 OpenCV 结合进行图像、视觉处理，如何传输、处理图片，如何进行视频流处理等，这些内容有助于读者开发会看的机器人。

第8章介绍了如何进行 ROS 机器人建模，从最基础的 URDF 概念到如何利用 Solidworks 快速建模。

第9章介绍了如何在 Gazebo 中进行机器人仿真，以及如何搭建仿真环境。

最后的附录介绍了与 Linux、ROS 相关的基本命令行工具，使缺乏基础的读者也能快速上手。

## 勘误和支持

由于笔者的水平有限，再加上编写时间仓促，书中难免会出现一些表达不尽人意甚至错误的地方，恳请读者批评指正（可以通过 [mzl626@163.com](mailto:mzl626@163.com) 与笔者进行交流）。期待能够得到大家的真挚反馈，在技术的道路上我们一起共勉。

## 致谢

感谢机械工业出版社华章公司的杨绣国编辑，在大半年的时间里她始终支持我的写作，并能允许我根据自己的时间安排写作计划。

感谢我的家人和孩子，他们一直尊重我的写作爱好，允许我把大量的业余时间（一部分本是用来陪伴他们的）花到写作上。

# Contents | 目 录

## 前言

## 第 1 章 机器人基础 ..... 1

- 1.1 机器人发展历史 ..... 1
- 1.2 ROS 发展历程 ..... 5
- 1.3 安装 ROS 环境 ..... 6

## 第 2 章 理解 ROS ..... 11

- 2.1 启动小海龟 ..... 11
- 2.2 系统架构 ..... 12
- 2.3 理解节点管理器 ..... 16
- 2.4 理解节点 ..... 19
- 2.5 理解话题 ..... 21
- 2.6 理解消息 ..... 23
- 2.7 理解服务 ..... 28
- 2.8 理解动作 ..... 30
- 2.9 理解参数 ..... 32

## 第 3 章 开发 ROS ..... 35

- 3.1 你好 ROS ..... 35
- 3.2 编译 ROS ..... 42
- 3.3 编译配置 ..... 49
- 3.4 调试 ROS ..... 62

## 第 4 章 资源与配置 ..... 67

- 4.1 消息通信 ..... 67

- 4.2 订阅服务 ..... 73

- 4.3 执行动作 ..... 77

- 4.4 启动管理 ..... 81

- 4.5 配置参数 ..... 89

- 4.6 分布式计算 ..... 97

## 第 5 章 运动控制 ..... 99

- 5.1 坐标系 ..... 99

- 5.1.1 理解坐标系 ..... 99

- 5.1.2 坐标变换 ..... 101

- 5.1.3 消息过滤 ..... 108

- 5.1.4 添加坐标系 ..... 111

- 5.2 控制器 ..... 113

- 5.2.1 控制器架构 ..... 114

- 5.2.2 双轮差速控制 ..... 121

## 第 6 章 语音处理 ..... 125

- 6.1 语音基础 ..... 127

- 6.2 语音识别 ..... 129

- 6.3 语音合成 ..... 137

- 6.4 声源定位 ..... 142

## 第 7 章 视觉处理 ..... 144

- 7.1 相机标定 ..... 144

- 7.2 图像处理 ..... 147

- 7.3 视频处理 ..... 152

<b>第 8 章 机器人建模</b> .....	157
8.1 URDF 模型基础 .....	157
8.2 实现 URDF 模型 .....	169
8.3 使用 Xacro 建模 .....	178
8.4 使用 Solidworks 建模 .....	186
<b>第 9 章 机器人仿真</b> .....	193
9.1 Gazebo 场景 .....	194

9.1.1 场景基础 .....	194
9.1.2 创建场景 .....	198
9.2 Gazebo 模型 .....	199
9.3 启动仿真 .....	207

<b>附录 A 常见 Linux、ROS 命令</b> .....	210
-----------------------------------	-----

<b>参考文献</b> .....	217
-------------------	-----



# 第 1 章

## 机器人基础

从春秋时期鲁班削竹木以为鹊，成而飞之，三日不下；到东汉末年毕岚制造了最早的水车“翻车”；再到三国时期蜀汉丞相诸葛亮制造了木牛流马，自动化工具和机器人一直承载着人们解放自身、提高效率的梦想。

随着人工智能和各类传感器技术的日趋成熟，目前机器人已不再局限于工业机器人领域，各种类型的服务机器人，如家庭机器人、教育机器人、无人机、迎宾机器人、物流机器人、厨房机器人、扫地机器人、客服机器人、医疗机器人等，如雨后春笋般不断涌现，承载着人类自动驾驶梦想的无人驾驶也进展迅速，整个社会正在逐步进入机器人时代。

就机器人操作系统而言，目前，市场上已经推出了多款可用的机器人操作系统（如阿里巴巴的 YunOS，Google 的 Android，Ubuntu，意大利的 YARP，美国的 Player Stage，日本的 Open Robot，等等），这其中最知名的便是 ROS。得益于 ROS 的灵活、低耦合、分布式和开源特性以及众多功能强大的可复用程序包，ROS 正在逐步成为机器人领域的事实标准。

### 1.1 机器人发展历史

1920 年，捷克作家 Karel Capek 在他的戏剧作品《罗素姆万能机器人》（Rossum's Universal Robots）中，根据 Robota（捷克文，原意为“劳役、苦工”）和 Robotnik（波兰文，原意为“工人”）的含义，创造出了机器人（Robot）这个词。

1942 年，美国科幻作家 Isaac Asimov 在一篇名为《环舞》（Runaround）的短篇小说中提出了著名的“机器人学三定律”。

第一定律：机器人不得伤害人类，或坐视人类受到伤害。

第二定律：除非违背第一定律，机器人必须服从人类的命令。

第三定律：在不违背第一及第二定律的情况下，机器人必须保护自己。

在 1956 年的达特茅斯会议（Dartmouth Conference）上，人工智能之父 Marvin Lee Minsky 提出了他对智能机器的定义，即智能机器“能够创建周围环境的抽象模型，如果遇

到问题，能够从抽象模型中寻找解决方法”。这一定义影响了以后 30 年智能机器人的研究方向。

### 1. 工业机器人

作为面向工业领域的多关节机械手或多自由度的机器装置，工业机器人能够自动执行任务，是靠自身动力和控制能力来实现各种功能的一种机器。工业机器人既可以接收人类的指令，也可以按照预先编排的程序来运行。它可以从事弧焊、点焊、码垛、装配、搬运、注塑、冲压、喷漆等多项工作。

1954 年，最早提出工业机器人概念的美国人 George Devol 针对他设计的机械手申请专利，这种机械手能够按照不同的程序从事不同的工作。1956 年，George Devol 与 Joseph Frederick Engelberger 联手制造出了第一台工业机器人 Unimate。随后，世界上第一家机器人制造企业 Unimation 公司成立，由于 Joseph Frederick Engelberger 在工业机器人领域的突出贡献，他被行业称为“工业机器人之父”。世界上第一台机器人如图 1-1 所示。



图 1-1 世界上第一台工业机器人

工业机器人经过多年的发展，市场格局相对稳定，瑞士 ABB、日本 FANUC、日本 YASKAWA、德国 KUKA 等行业巨头占据了行业领先地位，不得不提的是在 2016 年，美的集团以约 37 亿欧元的代价获得了 KUKA 的控股权。

我国工业机器人起步于 20 世纪 70 年代初期，到 2016 年，我国已经成为全球最大的工业机器人市场，代表厂商包括新松、埃夫特、新时达、埃斯顿等。

同样在 2016 年，东南大学自动化学院开设了全国首个机器人本科专业，拉开了我国机器人人才系统培养的序幕。

### 2. 服务机器人

截至目前，服务机器人仍没有一个严格的定义，从国际机器人联合会（International

Federation of Robotics, IFR) 的表述来看, 服务机器人是指通过半自主或完全自主运作, 为人类健康或设备良好状态提供帮助的机器人。从广泛意义上讲, 是否用于工业操作, 是区分工业机器人和服务机器人的重要特征, 不包含工业性操作的机器人均可称为服务机器人。

服务机器人的应用范围很广, 可以从事维护保养、修理、运输、清洗、保安、救援、监护、迎宾、交流、娱乐、业务办理等多项服务工作。

目前应用比较广泛的服务机器人包括扫地机器人、仓储机器人、迎宾机器人、餐饮机器人、政企机器人、陪伴机器人、导购机器人、娱乐机器人等。

在 1966 年到 1972 年之间, 美国国际斯坦福研究所 (SRI) 研制出机器人 Shakey。Shakey 带有视觉传感器, 能根据人的指令发现并抓取积木, 不过控制它的计算机有一个房间那么大, Shakey 被认为是世界第一台智能机器人。Shakey 智能机器人如图 1-2 所示。



图 1-2 Shakey 智能机器人

1969 年, 日本早稻田大学加藤一郎实验室研发出了第一台以双脚走路的机器人 Kirobo。加藤一郎由于较早进行并长期致力于研究仿人机器人, 因此被誉为“仿人机器人之父”。

1999 年, 日本索尼公司首次推出犬型机器人 AIBO, 当即销售一空, 从此娱乐机器人成为目前机器人迈进家庭的切入点之一。AIBO 犬型机器人如图 1-3 所示。

2002 年, 丹麦 iRobot 公司首次推出扫地机器人 Roomba, 它能避开障碍, 自动规划行进路线, 还能在电量不足时, 自动驶向充电座进行自主充电。从此机器人开始逐渐进入到普通民众的视野。Roomba 扫地机器人如图 1-4 所示。

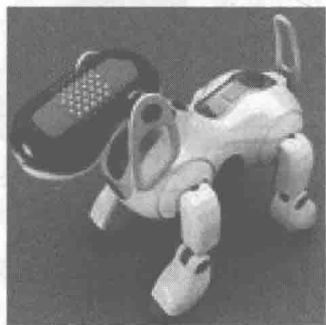


图 1-3 AIBO 犬型机器人



图 1-4 Roomba 扫地机器人

2006 年, 专注于无人飞行器控制系统及无人机解决方案的大疆诞生, 大疆无人机被广泛应用于航拍、遥感测绘、森林防火、电力巡线、搜索及救援、影视广告等工业及商

业用途，同时亦是全球众多航模航拍爱好者的最佳选择，大疆在无人机领域的持续努力已经使大疆成为我国为数不多的技术领先全球的企业之一。大疆无人机如图 1-5 所示。



图 1-5 大疆无人机

2015年由日本软银集团和法国 Aldebaran Robotics 研发的 Pepper 机器人正式发售，Pepper 可以综合考虑周围环境，并积极主动地做出反应。Pepper 配备了语音识别技术、呈现优美姿态的关节技术以及分析表情和声调的情绪识别技术，可与人类进行交流。Pepper 机器人如图 1-6 所示。

2018年7月，搭载基于百度 Apollo 平台的全球首款 L4 级量产自动驾驶巴士“阿波龙”量产下线。阿波龙前后安装有激光雷达、超声波雷达等传感器，能持续监测路面状况、周围物体，具有车流判断、路牌识别、避障等能力。阿波龙如图 1-7 所示。

我国在服务机器人领域的研发与日本、美国等国家相比起步较晚，但发展迅速，目前已与国外处于同一技术发展水平，代表厂商有哈工大、科沃斯、大疆等。

随着人工智能、移动互联网等技术的逐步成熟和中国制造 2025 的推进，服务机器人由于巨大的市场潜在规模，受到了诸多创业者的关注，并成为投资、融资领域的行业热点之一。

目前，我国服务机器人在部分细分市场（如无人机、电动平衡车、扫地机器人、搬运机器人、科教机器人等领域）已经处于全球领先地位；但受限于技术的成熟度，特别是高端核心部件、核心算法受制于人，以及服务机器人的产品痛点、形态、产业、规模、服务模式等仍需要验证，因此，我国的服务机器人行业整体仍处于探索发展期。

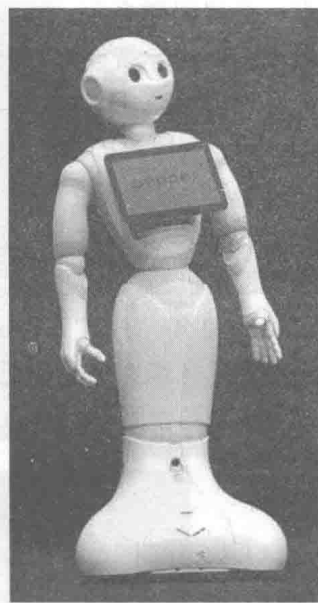


图 1-6 Pepper 机器人



图 1-7 阿波龙自动驾驶巴士

## 1.2 ROS 发展历程

ROS 作为目前最受欢迎的机器人操作系统，它的核心代码采用 C++ 编写并经由 BSD 许可发布，也正是基于此，本书采用 C++ 进行示例介绍。

ROS 源于 2007 年斯坦福大学和机器人技术公司 Willow Garage 合作项目 Switchyard。ROS 项目的最初愿景是提高机器人项目代码的复用率，构建一个能够整合不同研究成果、可重用代码的机器人软件平台。2012 年，ROS 团队从 Willow Garage 公司独立出来，并成立了一家非营利组织，即开源机器人基金会（Open Source Robotics Foundation, OSRF），ROS 自 2013 年起移交给该基金会管理。

目前 ROS 已经拥有数千个程序包，涵盖了硬件驱动、模拟仿真、运动规划、运动控制、环境感知等诸多领域，但真正的任务调度、编译、寻址等任务仍由底层的操作系统完成，这也是 ROS 被称为次级操作系统的原因。

基于 ROS 开发的著名机器人有 pr2、turtleBot、Husky、Erratic 等。turtleBot 机器人如图 1-8 所示。

自 2010 年 3 月推出第一个发行版 Box Turtle 以来，ROS 以每年 1~2 个版本的节奏已经发布了多个版本。ROS 作为次级操作系统，官方推荐基于 Ubuntu 操作系统运行，并跟随 Ubuntu 的发布节奏发布，在偶数年发布维护周期为 5 年的长期演进版本，在奇数年发布维护周期为 2 年的普通版本。



图 1-8 turtleBot 机器人

目前 ROS 最新版本为 2018 年 5 月发布的 Melodic Morenia，其官方支持时间为 5 年，上一个长期演进版本为 2016 年 5 月发布的 Kinetic Kame。ROS 已发布的历史版本如表 1-1 所示。

表 1-1 ROS 的历史版本

发行版本	发布日期	截止日期	Ubuntu 版本	备注
Melodic Morenia	2018 年 5 月	2023 年 5 月	18.04 LTS	LTS
Lunar Loggerhead	2017 年 5 月	2019 年 5 月	17.04	
Kinetic Kame	2016 年 5 月	2021 年 5 月	16.04 LTS	LTS
Jade Turtle	2015 年 5 月	2017 年 5 月	15.04	
Indigo Igloo	2014 年 7 月	2019 年 4 月	14.04 LTS	LTS
Hydro Medusa	2013 年 9 月	2015 年 5 月	13.04	
Groovy Galapagos	2012 年 12 月	2014 年 7 月	12.10	
Fuerte Turtle	2012 年 4 月		12.04 LTS	
Electric Emys	2011 年 8 月		11.10	
Diamondback	2011 年 3 月		11.04	
C Turtle	2010 年 8 月		10.04 LTS	
Box Turtle	2010 年 3 月		9.10	

ROS 的成功之处在于其有一个强大、灵活的模块化框架，不同的业务可以封装在模块中，业务实现与基础框架的解耦。ROS 吸收了很多其他开源项目的代码，如 Player 的驱动与运动控制及仿真、OpenCV 的视觉算法、OpenRAVE 的规划算法等。

不过由于 ROS 的基础架构设计于 2007 年，在应对机器人集群、实时性要求、非理想网络、可靠性要求很高的环境等复杂场景的需求时仍稍显不足，因此整合最新技术如 Zeroconf、Protocol Buffers、ZeroMQ、Redis、WebSockets、DDS 等的 ROS 2 正在紧锣密鼓的开发中。其中 ROS 2 的第一个版本 Ardent Apalone 在 2017 年 12 月正式发布，Ardent Apalone 全面支持 Ubuntu、Mac OS X、Win 10 等操作系统；取消了 Master 节点，实现了节点的分布式发现、发布 / 订阅、请求 / 响应、支持实时处理；并采用了新的编译系统 Ament。不过 ROS 2 仍需要完善，距离商用还有一段很长的路要走，而商用系统仍以 ROS 为主。

### 1.3 安装 ROS 环境

本书中采用的 ROS 版本为 ROS Melodic，操作系统为 Ubuntu 18.04 LTS 64bit。ROS Melodic 依赖的开发环境具体如下：

```
Ubuntu 17.10 及以上 (推荐 Ubuntu 18.04 LTS)
C++ 14
```

```

GCC 4.9
Python 2.7.* // 兼容 Python 3.5 及以上
LISP SBCL 1.3.14
CMake 3.10.2
Boost 1.65.1
Ogre 1.9
Gazebo 9.0.0
PCL 1.8.1
OpenCV 3.2
Qt 5.9.5
PyQt 5.10.1

```

需要注意的是，如果开发者准备采用 Python 进行开发，由于 Python 2.7 将在 2020 年停止支持，那么请使用 Python 3.5 进行。本书将采用 C++ 作为开发语言。

### 1. 准备环境

关于如何安装 Ubuntu 18.04 LTS，请开发者通过“参考文献 [3]”自行学习，本书中将不做介绍，推荐安装 Ubuntu 18.04 LTS 的磁盘空间至少为 50G。

需要特别说明的是，如果开发者的 PC 上已经安装了 Windows 操作系统，那么考虑到采用虚拟机的模式安装 Ubuntu 18.04 LTS 会影响 ROS 的运行性能，因此推荐采用双系统的模式安装 Ubuntu 18.04 LTS。另外推荐将 Ubuntu 18.04 LTS 的更新源切换为国内更新源 <http://mirrors.aliyun.com/ubuntu>，下载速度会较快。

对于 Ubuntu 18.04 LTS，在如图 1-9 所示的界面中点击“下载自”下拉菜单即可更换安装源。

若读者希望在 Ubuntu 16.04 LTS 上安装 ROS Kinetic，考虑到 ROS Kinetic 依赖于 Python 2.7，那么为了保证 ROS 安装的顺利进行，可利用 Ubuntu 系统中专门维护系统命令链接符的工具 `update-alternatives` 检查 Python 的配置，方法如下：

```
$sudo update-alternatives --config python
```

输出结果如下：

```

有 2 个候选项可用于替换 Python (提供 /usr/bin/python)。
  选择          路径                优先级  状态
-----
0              /usr/bin/python3      300     自动模式
* 1              /usr/bin/python2      100     手动模式
2              /usr/bin/python3      300     手动模式

```

要维持当前值 [\*] 请按 < 回车键 >，或者键入选择的编号：

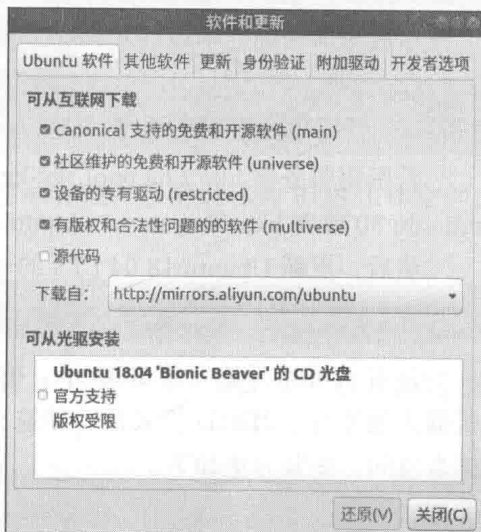


图 1-9 软件及更新

默认情况下，Python 对应的软连接为 /usr/bin/python2，如果为 /usr/bin/python3，那么请按上述提示调整为 /usr/bin/python2，在如上环境中，输入编号“1”。

## 2. 安装 ROS

为了安装 ROS Melodic，首先需要在 Ubuntu 18.04 LTS 上添加安装源到 sources.list，方法如下：

```
$sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

另外 ROS 提供了国内的安装源，采用国内的安装源下载速度会更快，添加国内中国科学技术大学的安装源的示例代码如下：

```
$sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.ustc.edu.cn/ros/ubuntu/ $DISTRIB_CODENAME main" > /etc/apt/sources.list.d/ros-latest.list'
```

然后添加密钥到本地的 trusted 数据库中并加以验证，方法如下：

```
$sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
```

若密钥服务器 hkp://ha.pool.sks-keyservers.net:80 无法添加成功，则可以尝试 hkp://pgp.mit.edu:80 或者 hkp://keyserver.ubuntu.com:80。

然后，更新 Ubuntu 18.04 LTS 的可用软件包列表，方法如下：

```
$sudo apt-get update
```

接着就可以开始安装 ROS 了，推荐安装桌面完整版，这样就可以将 ROS、rqt、rviz、机器人通用库、2D/3D 仿真器、导航、2D/3D 感知等功能一次性安装完成，这需要较大的磁盘空间，安装方法如下：

```
$sudo apt-get install ros-melodic-desktop-full
```

其中，rqt 是一个用来为 ROS 机器人开发图形接口的 QT 框架，常用于观察 ROS 节点当前的信息，如警告信息、位置信息和话题的信息等；rviz 则是一个非常强大的 3D 可视化模拟环境，支持模拟多种传感器数据和基于 URDF 模型的机器人模拟。

如果希望安装特定的 ROS 程序包，则可以首先查看当前可用的程序包列表，然后选择自己需要的程序包进行安装，示例代码如下：

```
$apt-cache search ros-melodic
$sudo apt-get install ros-melodic-slam-gmapping
```

在开始使用 ROS 之前，还需要安装 rosdep，rosdep 是 ROS 一些核心组件的运行基础，会在编译 ROS 源码时检查和安装程序包依赖，如果 rosdep 没有正常安装，那么 ROS 的部分工具将会出现异常。安装 rosdep 的方法如下：

```
$sudo rosdep init // 初始化
$rosdep update // 配置账户，在当前用户目录下生成 .ros 目录
```



默认情况下，ROS\_HOME 的路径为“~/ros”，该目录下保存着日志文件和测试结果文件。另外，在第一次启动 ROS 时，需要进行 ROS 环境变量配置，环境变量配置如下：

```
$echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$source ~/.bashrc
```

查看环境变量是否设置正确，方法如下：

```
$sudo printenv | grep ROS
```

在笔者的电脑上，查看环境变量的结果如下：

```
ROS_ROOT=/opt/ros/melodic/share/ros
ROS_PACKAGE_PATH=/opt/ros/melodic/share
LD_LIBRARY_PATH=/opt/ros/melodic/lib
PATH=/opt/ros/melodic/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
PYTHONPATH=/opt/ros/melodic/lib/python2.7/dist-packages
PKG_CONFIG_PATH=/opt/ros/melodic/lib/pkgconfig
CMAKE_PREFIX_PATH=/opt/ros/melodic
ROS_ETC_DIR=/opt/ros/melodic/etc/ros
```

至此，ROS 的核心程序包就安装好了，为了构建和管理开发者自己的 ROS 工作空间，还需要安装 rosinstall，安装 rosinstall 的方法如下：

```
$sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

成功安装 rosinstall 之后，就可以体验 ROS 和创建 ROS 应用了。如果希望基于 ROS 源代码进行学习，则可以下载完整的 ROS 源代码，方法如下：

```
$sudo apt-get install python-rosdep python-rosinstall-generator python-wstool
python-rosinstall build-essential
$mkdir ~/ros_catkin_ws
$cd ~/ros_catkin_ws
$rosinstall_generator desktop_full --rostdistro melodic --deps --tar > melodic-
desktop-full.rosinstall
$wstool init -j8 src melodic-desktop-full.rosinstall
```

如果希望在网络发生中断后可以以断点续传的方式继续下载，那么命令如下：

```
$wstool update -j 8 -t src
```

ROS 将使用 wstool 进行版本管理。关于基于 ROS 源代码如何进行编译的相关内容，本书将不做介绍。

如何通过 ROS 源代码学习 ROS 的更多实现细节，可以在 Ubuntu 18.04 LTS 下通过 opengrok 进行浏览，opengrok 可以通过 Ubuntu 18.04 LTS 的软件管理器直接进行安装。

### 3. Qt Creator 开发环境

ROS 支持 C++、Python、Lisp、node.js 等多种开发语言，开发者可以基于 Eclipse 利