



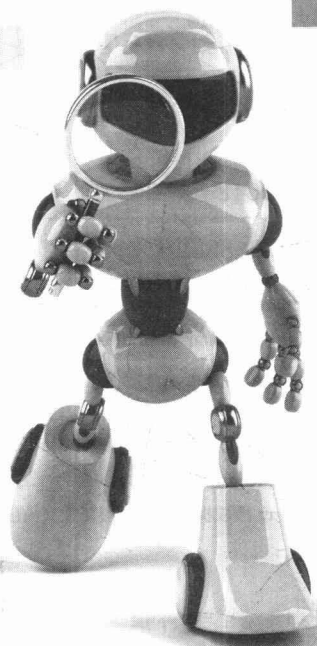
ROS机器人开发 实用案例分析

ROS Robotics By Example

[美] 卡罗尔·费尔柴尔德 (Carol Fairchild) 托马斯 L.哈曼 (Thomas L. Harman) 著
吴中红 石章松 潘丽 等译
刘锦涛 审校



机械工业出版社
China Machine Press



ROS机器人开发

实用案例

ROS Robotics By Example

[美] 卡罗尔·费尔柴尔德 (Carol Fairchild) 托马斯 L. 哈曼 (Thomas L. Harman) 著

吴中红 石章松 潘丽 等译

刘锦涛 审校



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

ROS 机器人开发: 实用案例分析 / (美) 卡罗尔·费尔柴尔德 (Carol Fairchild), (美) 托马斯 L. 哈曼 (Thomas L. Harman) 著; 吴中红等译. —北京: 机械工业出版社, 2018.3
(机器人设计与制作系列)

书名原文: ROS Robotics By Example

ISBN 978-7-111-59372-0

I. R… II. ①卡… ②托… ③吴… III. 机器人—程序设计 IV. TP242

中国版本图书馆 CIP 数据核字 (2018) 第 048798 号

本书版权登记号: 图字 01-2016-8652

Carol Fairchild, Thomas L. Harman: ROS Robotics By Example (ISBN: 978-1-78217-519-3).

Copyright © 2016 Packt Publishing. First published in the English language under the title “ROS Robotics By Example”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2018 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

ROS 机器人开发: 实用案例分析

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: 张锡鹏

责任校对: 李秋荣

印 刷: 三河市宏图印务有限公司

版 次: 2018 年 4 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 20

书 号: ISBN 978-7-111-59372-0

定 价: 79.00 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

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

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

工欲善其事，必先利其器。作为一门实践性极强的交叉性学科，机器人学的学习过程要求学习者结合各类机器人进行实践操作，来学习相关应用程序的开发。然而，各类机器人平台的价格令许多学习者望而却步。

目前市面上诸多介绍机器人开发的书籍中，有的专注于理论讲解，其中一串串的复杂公式令人望而生畏，有的虽然主要介绍实践性的内容，但忽略了很多读者没有一台真正的机器人的现状。而一台机器人往往价值不菲，面对充满诱惑的机器人技术学习之旅，许多人因为没有一台机器人而戛然而止。本书将为那些没有实际机器人的爱好者提供一条学习机器人应用程序开发的有效途径。

本书针对基于 ROS 的机器人开发技术进行介绍。ROS 的特点和优点在许多书籍中都做过介绍，这里不再赘述。下面着重对本书的特点进行说明，以利于读者选择和学习。

- 涵盖模拟机器人模型构建与真实机器人操控，有无真机均可上手。

本书对使用 URDF 文件以及 SDF 文件来构建模拟的两轮机器人、机器人手臂模型的方法进行了详细的描述，并对 Gazebo 仿真系统知识进行了深入的讲述，为读者构建自己的模拟机器人提供了引导，同时为没有实际机器人的读者提供了学习机器人开发的途径。

- 机器人类型包括地面机器人、空中机器人与机器人手臂，全面满足各类读者。

本书对 TurtleBot 机器人、Bebop 四旋翼无人机、Crazyflie 四旋翼无人机以及 Baxter 机器人手臂均进行了全面的介绍，内容包括各类机器人的虚拟模型构建、导航控制、机械臂的关节控制、路径规划、基于外部设备（游戏手柄、控制开发板等）的机器人控制以及 ROS 的高级应用等内容，内容全面而新颖。

- 应用场景包括机器人操控、导航以及机器人手臂路径规划、移动应用开发。

针对各类机器人，分别针对相应类型机器人的特点与主要应用场景，进行了模拟环境下的模型构建与操控以及典型真实机器人操控的内容讲解，并提供了相应的源代码，以供参考和学习，极大地降低了读者的学习难度，便于与作者提供的例程进行对照学习，事半功倍。

本书的两位作者均是长期从事 ROS 机器人研究与教学的专家，本书的内容深入浅出、

覆盖面广，相信必将给读者带来不一样的学习体验。尽管目前 ROS 版本已经发展到 Lunar，但事实上，Indigo 仍是目前应用最为广泛的发行版，也是机器人研究领域开源项目支持最多的版本。相较于目前的 Jade、Kinetic 以及 Lunar 版本，Indigo 版本并无本质上的区别，其区别主要体现在与 Ubuntu 系统（Linux 系统）内核版本的绑定上。而随着 ROS 2.0 的发布，在实时处理方面有需要的读者则可以直接考虑学习 ROS 2.0，而本书也同样能够提供一定的参考。

鉴于译者水平有限，书中难免存在不足与纰漏，欢迎读者不吝赐教，对于书中的不当之处，读者可以发送邮件到 yizhousan@163.com，感谢您的来信与指正。

About the Authors | 作者简介

卡罗尔·费尔柴尔德 (Carol Fairchild) 是 Fairchild Robotics 公司的所有人、首席工程师，该公司致力于机器人技术的开发与集成。卡罗尔是休斯顿大学明湖分校 (University of Houston-Clear Lake, UHCL) Baxter 实验室的一名研究人员，同时也是一名兼职教授。她的研究领域主要是基于 Baxter 机器人的扩展应用开发。在费尔柴尔德女士早期开始建造她的第一台机器人 (这台机器人被称为 Heathkit Hero) 时，她就已经开始了机器人技术方方面面的学习与实践。她在得州农工大学 (Texas A&M) 获得工程技术学士学位，并在休斯顿大学明湖分校获得计算机工程的硕士学位。费尔柴尔德女士曾在中学教授机器人学课程，是第一乐高联盟 (FIRST LEGO League, FLL) 的教练，同时也是灵感和科技大赛 (For Inspiration and Recognition of Science and Technology, FIRST) 的志愿者。

托马斯 L. 哈曼 (Thomas L. Harman) 博士是休斯顿大学明湖分校工程系的主任。他的研究领域主要是控制系统、机器人技术与微处理器技术的应用。他和他同事关于机器人和激光的论文已经应用在了医学领域。2005 年，他当选为休斯顿大学明湖分校的特聘教授。他曾担任 FIRST 机器人大赛的裁判与安全顾问。托马斯博士已经出版的专著、合著书籍有 18 本，主题包括微处理器技术、MATLAB 与仿真以及美国国家电气规程等。他在休斯顿大学明湖分校的实验室有一台 Baxter 双臂机器人、多台 TurtleBot 机器人以及其他几款机器人。

前言 | Preface

学习 ROS，并通过支持 ROS 的机器人（如 Baxter 和 TurtleBot）进行实际操作，将开启一段神奇的机器人技术的学习之旅。ROS 具有十分突出的优点以及鲜明的特色，但其学习曲线也较为陡峭。通过对许多实际的 ROS 应用的尝试，并在总结了一系列错误的经验教训之后，我们摸索出了一条切实有效的学习路径，并形成了本书所论述的内容。本书会呈现我们关于 ROS 的见解中最重要的知识，并在读者的 ROS 学习之旅中提供循序渐进的引导。我们将结合典型的 ROS 机器人进行介绍，这些机器人包括 TurtleBot、Baxter、Crazyflie 以及 Bebop，此外还将通过模拟机器人对相关技术进行论述，这些模拟机器人主要有 Turtlesim 和 Hector。

除了对一般性的 ROS 技术进行论述外，本书还将对 ROS 机器人的高级应用特性进行讲解。本书首先对 ROS 的基本配置进行介绍，内容包括 ROS 的安装、基本概念、主要的功能包与工具简介等。然后对相应的故障排查方法进行了论述，当读者在实际操作中得到的结果与预期结果不相符时，可以按照相应方法进行故障的排查。然后通过模拟的方式，先对 Turtlesim 的 ROS 组成模块进行了描述，之后对其他几类典型机器人的 ROS 组成模块进行了相应的介绍。对 ROS 组成模块的介绍，一般遵循的流程是首先介绍基本的 ROS 命令，然后对 ROS 的功能包、节点、主题以及消息进行论述，以此来对 ROS 机器人操作系统的整体内容进行详细的阐述。为了对书中选用的机器人的整体性能进行描述，书中还给出了相关机器人的技术资料。

ROS 涵盖了完整的软件概念、实现以及相应的工具，其目标是为机器人提供复杂系统的同质化图景以及一体化的软件集成。传感器的外部扩展库、执行器驱动以及相应的接口均已包含其中，此外 ROS 还囊括了大部分目前最新最有效的算法。对于如 OpenCV 这样十分流行的第三方开源项目，ROS 并未包含，但可以直接对其进行导入。此外，为了节省研究人员的时间，使研究人员能够将精力集中于自己的兴趣点上，ROS 还提供了一系列的辅助工具，来对机器人进行控制、监控以及调试，这些工具包括 rqt、rviz、Gazebo、动态配置工具以及 MoveIt 等。

本书将循序渐进地以实例的形式对 ROS 的各部分内容进行阐述，每一部分内容都是实

际的机器人实例的组成部分。对于 TurtleBot 机器人，主要对导航和地图构建主题进行探索；对于 Baxter 机器人，则主要介绍关节控制与路径规划主题。对于书中提及的机器人，将以简单的 Python 脚本为例来实现 ROS 的各个元素。读者都能够通过模拟机器人对书中内容加以实操练习。此外，书中还介绍了如何以模拟的形式设计并构建读者自己的机器人模型。

ROS 本身的优异性能、支持 ROS 的机器人的多样性，以及 ROS 社区的支持，共同构建起了一个值得一试的机器人开发 ROS 实例的学习之旅。除了书中的内容之外，读者还能够找到丰富的在线扩展指南、wiki 资源、论坛、开发技巧等内容。那么现在就从本书开始，一起踏上 ROS 机器人开发的学习之旅吧。

本书涵盖的内容

第 1 章主要阐述了学习 ROS 的优势，同时论述了支持 ROS 的机器人平台的主要亮点。对 ROS 在 Ubuntu 操作系统下的安装流程进行了介绍。描述了 ROS 的框架结构，并对其主要组件进行了描述。对 Turtlesim 模拟机器人程序进行了简述，同时为了进一步加深读者对 ROS 组件以及 ROS 命令的理解，对 Turtlesim 的使用进行了介绍。

第 2 章主要介绍了 ROS 的模拟环境 Gazebo。本章内容将引导读者一步步构建起一台差分驱动的两轮模拟机器人，并介绍了统一机器人描述格式（Universal Robotic Description Format, URDF）的相关内容。此外还将详细地介绍基本的 ROS 工具（如 rviz、Gazebo 等）的使用方法，从而使读者能够在模拟环境下展示所构建的机器人，并与之进行交互。

第 3 章主要介绍一款真实的机器人——TurtleBot 的操作和控制。如果读者没有该型机器人，那么可以在 Gazebo 模拟环境下使用模拟的 TurtleBot 机器人来学习本章的内容。读者可以通过 ROS 命令、Python 脚本程序等多种方法实现对 TurtleBot 机器人的操控。此外，本章还讲述了使用 rqt 工具及其插件对 TurtleBot 进行控制以及对其传感器数据进行监控的方法。

第 4 章探讨了视觉传感器以及对于机器人创建周围环境地图的相关内容。论述了可用作 TurtleBot 视觉系统的 3D 传感器，并介绍了在 ROS 下进行设置与操作，使得机器人具备自主导航能力的内容。在 TurtleBot 机器人导航模块中，应用了同步定位与地图创建技术。

第 5 章对复杂的机器人手臂进行了简要介绍。采用 Xacro 的宏语言设计并构建了一个模拟的机器人手臂。为了操作机器人手臂，在 Gazebo 环境下创建了相应的控制器。通过控制器的创建，论述了简单的机器人手臂的结构和物理学的相关内容。

第 6 章对复杂的机器人手臂控制进行了更进一步的探讨。对最新型的 Baxter 机器人进

行了介绍，这也是本书中介绍的第二款机器人。Baxter 机器人有两个 7 自由度的手臂以及其他一系列的传感器，并提供了一个开源的 Baxter 机器人模拟器，本章正是采用这一机器人手臂模拟器进行论述。通过实例的形式，分别在位置控制模式、速度控制模式与扭矩控制模式下，对机器人手臂的正向与反向运动控制进行了介绍。此外，本章还对一个非常有用的 ROS 工具——MoveIt 进行了介绍，MoveIt 既可以对模拟的 Baxter 机器人手臂进行运动规划的仿真和执行，也可以对真实的 Baxter 机器人手臂进行运动规划的仿真和执行。

第 7 章对一个日渐风靡的机器人领域——无人机进行了论述。本章主要针对四旋翼的相关内容进行阐述，对四旋翼硬件与飞行控制的概念进行了论述。对四旋翼模拟器 Hector 的下载与控制进行了介绍，通过对模拟四旋翼飞行操控的练习，使读者能够对真实的四旋翼（如 Bitcraze 的 Crazyflie 或者 Parrot 的 Bebop）进行操控。对四旋翼的控制可以使用遥控或者 ROS 的主题 / 消息命令实现。

第 8 章对一系列可以用来对支持 ROS 的机器人进行控制的外部设备进行了描述。Joystick 游戏手柄、控制器开发板（Arduino 和 树莓派）以及所有移动设备均具有 ROS 接口，这些接口可以集成到读者的机器人系统之中来实现对机器人更多的外部控制。

第 9 章对书中前面章节所介绍的多个 ROS 组件与概念进行集成，从而实现一个十分具有挑战性的任务：自主飞行。在该任务中，读者将在采用 Kinect 3D 传感器构建的地图下控制 Crazyflie 四旋翼飞行器自动飞往一个“远方”的目标。这一任务采用了 ROS 的消息通信机制与协同传输，从而得到四旋翼飞行器上 Kinect 的视野以及目标相对于四旋翼的位置，从而对飞行进行精密的规划，在此基础上，将采用了 PID 控制的 Crazyflie 的飞行控制软件作为飞行任务中的飞行控制模块，从而实现了四旋翼的自主飞行。

第 10 章对高级 ROS 机器人应用的相关内容进行了论述，激励读者对 ROS 功能做更加深入的学习。对机器人的语音控制、人脸检测与识别以及机器人语音功能等都做了描述。

学习本书内容所需的基础知识与设备

本书的形式是给出各项任务内容的步骤指令，并由读者参与其中，具体实操。读者需要一台安装了 Ubuntu 14.04 (Trusty Tahr) 的电脑。同时也支持其他版本的 Ubuntu 以及 Linux 发行版，此外也支持 Mac OS X、Android 和 Windows，不过需要参考 ROS wiki 的文档（地址：<http://wiki.ros.org/Distributions>）。

本书中所用的 ROS 版本为 Indigo Igloo，该版本是目前推荐的稳定发行版，对该版本的支持将持续到 2019 年 4 月。书中内容对其他版本的 ROS 理论上是可支持的，但是并没有进行测试。

书中所有的软件都是开源的，可以免费获取和使用。在对相关软件进行介绍时会提供相关的下载说明。本书第 1 章给出了下载和设置 ROS 软件环境的相应指令。

下载软件时，推荐使用 Debian 软件包。当不存在相应的 Debian 包时，推荐从 GitHub 软件发行库中下载。

Gazebo 模拟器提供了增强的图形化处理功能，推荐使用专用的图形选项卡进行操作，但并不强求。

涉及的外部设备主要包括 3D 传感器、Xbox 或者 PS3 控制器，以及 Arduino 或者是 Raspberry Pi 控制板、Android 移动设备等。

本书面向的读者人群

本书主要面向机器人开发人员，无论是业余爱好者还是相关研究人员，无论是专业人士还是对 ROS 感兴趣的人，都能从书中获益，如果具备 GNU/Linux 系统和 Python 语言的相关知识则更好。

示例代码下载

读者也可以在 GitHub 上下载相应的代码包，地址为：<https://github.com/Packt Publishing/ROS-Robotics-By-Example>。此外，我们也给出了丰富的其他各类书籍的相关下载内容（代码包、视频等），地址为：<https://github.com/Packt Publishing/>，欢迎读者下载使用。

书中彩图下载

我们还以 PDF 文件格式的方式提供了书中有关的屏幕截图以及图表的彩图文件。这些彩图能够便于读者理解输出内容的变化。读者可以通过下述网址下载相关内容：http://www.packtpub.com/sites/default/files/downloads/ROSRoboticsByExample_Color Images.pdf。

目 录 | Contents

译者序

作者简介

前言

第 1 章 ROS 初体验 1

- 1.1 ROS 的用途以及学习 ROS 的好处 1
- 1.2 哪些机器人采用了 ROS 2
- 1.3 安装并启动 ROS 4
 - 1.3.1 配置 Ubuntu 系统的软件源 4
 - 1.3.2 设置 Ubuntu 系统软件源列表 5
 - 1.3.3 设置 Ubuntu 系统密钥 5
 - 1.3.4 安装 ROS Indigo 5
 - 1.3.5 初始化 rosdep 6
 - 1.3.6 环境设置 6
 - 1.3.7 安装 rosinstall 7
 - 1.3.8 故障排除——ROS 环境测试 7
- 1.4 生成一个 catkin 工作空间 7
- 1.5 ROS 的功能包与清单 8
 - 1.5.1 ROS 清单 9
 - 1.5.2 探索 ROS 功能包 9
- 1.6 ROS 节点与 ROS 节点管理器 11
 - 1.6.1 ROS 节点 11
 - 1.6.2 ROS 节点管理器 12

1.6.3 确定节点和主题的 ROS

命令 14

1.7 第一个 ROS 机器人模拟程序——

Turtlesim 15

1.7.1 启动 Turtlesim 节点 15

1.7.2 Turtlesim 节点 16

1.7.3 Turtlesim 主题与消息 18

1.7.4 Turtlesim 的参数服务器 20

1.7.5 移动乌龟的 ROS 服务 22

1.8 ROS 命令小结 23

1.9 本章小结 24

第 2 章 构建一个模拟的两轮

ROS 机器人 25

2.1 rviz 25

2.1.1 安装和启动 rviz 26

2.1.2 使用 rviz 27

2.2 生成并构建 ROS 功能包 29

2.3 构建差分驱动的机器人 URDF 30

2.3.1 生成机器人底座 31

2.3.2 使用 roslaunch 32

2.3.3 添加轮子 35

2.3.4 添加小脚轮 37

2.3.5 添加颜色 38

2.3.6 添加碰撞属性 40

2.3.7 移动轮子 41

2.3.8 tf 和 robot_state_publisher

简介	42	3.5.7 排查网络联接中的故障	70
2.3.9 添加物理学属性	42	3.5.8 TurtleBot 机器人系统 测试	70
2.3.10 试用 URDF 工具	43	3.6 TurtleBot 机器人的硬件规格 参数	72
2.4 Gazebo	45	3.7 移动真实的 TurtleBot 机器人	73
2.4.1 安装并启动 Gazebo	45	3.7.1 采用键盘远程控制 TurtleBot 机器人移动	74
2.4.2 使用 roslaunch 启动 Gazebo	46	3.7.2 采用 ROS 命令控制 TurtleBot 机器人移动	75
2.4.3 使用 Gazebo	47	3.7.3 编写第一个 Python 脚本 程序控制 TurtleBot 机器人移动	76
2.4.4 机器人 URDF 的修改	50	3.8 rqt 工具简介	79
2.4.5 Gazebo 模型验证	51	3.8.1 rqt_graph	79
2.4.6 在 Gazebo 中查看 URDF	51	3.8.2 rqt 的消息发布与主题 监控	82
2.4.7 机器人模型调整	53	3.9 TurtleBot 机器人的里程计	84
2.4.8 移动机器人模型	53	3.9.1 模拟的 TurtleBot 机器人 的测程	84
2.4.9 其他的机器人仿真环境	54	3.9.2 真实的 TurtleBot 机器人 的里程计在 rviz 下的显示	86
2.5 本章小结	55	3.10 TurtleBot 机器人的自动充电	88
第 3 章 TurtleBot 机器人操控	56	3.11 本章小结	90
3.1 TurtleBot 机器人简介	56	第 4 章 TurtleBot 机器人导航	91
3.2 下载 TurtleBot 模拟器软件	57	4.1 TurtleBot 机器人的 3D 视觉系统	92
3.3 在 Gazebo 中启动 TurtleBot 模拟器	58	4.1.1 3D 视觉传感器原理	92
3.3.1 常见问题与故障排除	60	4.1.2 3D 传感器对比	92
3.3.2 ROS 命令与 Gazebo	61	4.1.3 障碍物规避的缺陷	96
3.3.3 模拟环境下使用键盘 远程控制 TurtleBot	63	4.2 配置 TurtleBot 机器人并安装 3D 传感器软件	96
3.4 控制一台真正的 TurtleBot 机器人的准备	64	4.2.1 Kinect	96
3.5 联接上网本与远程计算机	66	4.2.2 ASUS 与 PrimeSense	98
3.5.1 网络类型	67		
3.5.2 网络地址	67		
3.5.3 远程计算机网络设置	68		
3.5.4 上网本网络设置	69		
3.5.5 安全外壳协议联接	69		
3.5.6 网络设置小结	70		

4.2.3	摄像头软件结构	98
4.2.4	术语界定	98
4.3	独立模式下测试 3D 传感器	99
4.4	运行 ROS 可视化节点	100
4.4.1	使用 Image Viewer	
	可视化数据	100
4.4.2	使用 rviz 可视化数据	102
4.5	TurtleBot 机器人导航	105
4.5.1	采用 TurtleBot 机器人	
	构建房间地图	105
4.5.2	采用 TurtleBot 机器人	
	实现自主导航	109
4.5.3	rqt_reconfigure	116
4.5.4	进一步探索 ROS 导航	117
4.6	本章小结	117

第 5 章 构建模拟的机器人手臂 119

5.1	Xacro 的特点	119
5.2	采用 Xacro 建立一个关节式	
	机器人手臂 URDF	121
5.2.1	使用 Xacro 属性标签	121
5.2.2	使用 roslaunch 启动	
	rrbot	124
5.2.3	使用 Xacro 的包含与	
	宏标签	126
5.2.4	给机器人手臂添加网格	129
5.3	在 Gazebo 中控制关节式机器人	
	手臂	133
5.3.1	添加 Gazebo 特定的元素	133
5.3.2	将机器人手臂固定在世界	
	坐标系下	135
5.3.3	在 Gazebo 中查看机器人	
	手臂	135
5.3.4	给 Xacro 添加控件	136

5.3.5	采用 ROS 命令行控制	
	机器人手臂	140
5.3.6	采用 rqt 控制机器人手臂	141
5.4	本章小结	143

第 6 章 机器人手臂摇摆的 关节控制 144

6.1	Baxter 简介	145
6.1.1	Baxter, 一款研究型	
	机器人	146
6.1.2	Baxter 模拟器	147
6.2	Baxter 的手臂	147
6.2.1	Baxter 的俯仰关节	149
6.2.2	Baxter 的滚转关节	149
6.2.3	Baxter 的坐标系	149
6.2.4	Baxter 手臂的控制模式	150
6.2.5	Baxter 手臂抓手	151
6.2.6	Baxter 手臂的传感器	152
6.3	下载 Baxter 软件	152
6.3.1	安装 Baxter SDK 软件	152
6.3.2	安装 Baxter 模拟器	154
6.3.3	配置 Baxter shell	155
6.3.4	安装 MoveIt	156
6.4	在 Gazebo 中启动 Baxter	
	模拟器	157
6.4.1	启动 Baxter 模拟器	157
6.4.2	“热身”练习	161
6.4.3	弯曲 Baxter 手臂	163
6.4.4	Baxter 手臂控制器的	
	调校	173
6.5	Baxter 手臂与正向运动学	174
6.5.1	关节与关节状态发布器	174
6.5.2	理解 tf	177
6.5.3	rviz 下的 tf 坐标系	180

6.5.4	查看机器人元素的 tf 树	181
6.6	MoveIt 简介	182
6.6.1	采用 MoveIt 给 Baxter 手臂进行运动规划	184
6.6.2	在场景中添加物体	185
6.6.3	采用 MoveIt 进行避障运动规划	186
6.7	配置真实的 Baxter 机器人	188
6.8	控制真实的 Baxter 机器人	190
6.8.1	控制关节到达航路点	190
6.8.2	控制关节的力矩弹簧	191
6.8.3	关节速度控制演示	192
6.8.4	其他示例	192
6.8.5	视觉伺服和抓握	192
6.9	反向运动学	193
6.10	本章小结	196

第 7 章 空中机器人基本操控 198

7.1	四旋翼飞行器简介	199
7.1.1	风靡的四旋翼飞行器	199
7.1.2	滚转角、俯仰角与偏航角	200
7.1.3	四旋翼飞行器原理	201
7.1.4	四旋翼飞行器的组成	203
7.1.5	添加传感器	203
7.1.6	四旋翼飞行器的通信	204
7.2	四旋翼飞行器的传感器	204
7.2.1	惯性测量单元	205
7.2.2	四旋翼飞行器状态传感器	205
7.3	放飞前的准备工作	205
7.3.1	四旋翼飞行器检测	206
7.3.2	飞行前检测列表	206
7.3.3	飞行中的注意事项	207

7.3.4	需要遵循的规则和条例	207
7.4	在无人机中使用 ROS	208
7.5	Hector 四旋翼飞行器简介	208
7.5.1	下载 Hector Quadrotor 功能包	209
7.5.2	在 Gazebo 中启动 Hector 四旋翼飞行器	210
7.6	Crazyflie 2.0 简介	216
7.6.1	无 ROS 情况下的 Crazyflie 控制	218
7.6.2	使用 Crazyradio PA 进行通信	218
7.6.3	加载 Crazyflie ROS 软件	219
7.6.4	放飞前的检查	222
7.6.5	使用 teleop 操控 Crazyflie 飞行	222
7.6.6	在运动捕获系统下飞行	226
7.6.7	控制多个 Crazyflie 飞行	226
7.7	Bebop 简介	227
7.7.1	加载 bebop_autonomy 软件	228
7.7.2	Bebop 飞行前的准备	229
7.7.3	使用命令控制 Bebot 飞行	230
7.8	本章小结	231

第 8 章 使用外部设备控制机器人 233

8.1	创建自定义 ROS 游戏控制器接口	233
8.1.1	测试游戏控制器	234
8.1.2	使用 joy ROS 功能包	236
8.1.3	使用自定义游戏控制器接口控制 Turtlesim	237

8.2 创建自定义 ROS Android 设备接口.....	242	9.2.3 使用 iai_kinect2 元包.....	271
8.2.1 使用 Turtlebot Remocon 进行操控.....	242	9.3 任务设置.....	277
8.2.2 使用 Android 设备实现 ROS 机器人的自定义控制.....	245	9.3.1 探测 Crazyflie 与目标.....	277
8.3 在 Arduino 或树莓派上创建 ROS 节点.....	249	9.3.2 使用 Kinect 与 OpenCV.....	281
8.3.1 使用 Arduino.....	249	9.3.3 对 Crazyflie 进行跟踪.....	283
8.3.2 使用树莓派.....	260	9.4 Crazyflie 控制.....	285
8.4 本章小结.....	261	9.5 放飞 Crazyflie.....	290
9.5.1 悬停.....	290	9.5.2 飞往静止目标.....	292
9.5.3 吸取的经验.....	294	9.6 本章小结.....	295
第 9 章 操控 Crazyflie 执行飞行任务.....	262	第 10 章 ROS 功能扩展.....	296
9.1 执行任务所需的组件.....	263	10.1 通过声音控制机器人.....	296
9.1.1 用于 Windows 的 Kinect v2.....	263	10.1.1 Sphinx 库.....	297
9.1.2 Crazyflie 操作.....	263	10.1.2 Rospeex 库.....	298
9.1.3 任务软件结构.....	264	10.2 给机器人添加语音功能.....	299
9.1.4 OpenCV 与 ROS.....	265	10.3 给机器人添加人脸识别功能.....	299
9.2 安装任务所需的软件.....	266	10.3.1 采用级联分类器进行人脸识别.....	300
9.2.1 安装 libfreenect2.....	267	10.3.2 采用 OpenCV 进行人脸识别.....	301
9.2.2 安装 iai_kinect2.....	269	10.4 本章小结.....	303

第 1 章

ROS 初体验

本章，我们将对机器人操作系统（Robot Operating System, ROS）进行介绍。ROS 是一系列软件的集合，主要用来辅助研究人员和开发人员进行机器人系统相关的研究和开发。我们将首先对 ROS 在 Ubuntu 系统下的安装方法进行介绍，在此基础上将对 ROS 的结构以及诸多组件进行论述。这些内容将有助于读者深入理解怎样使用 ROS 进行机器人应用软件的开发。

在对 ROS 进行介绍时，将分别对其主要元素与对应的功能进行介绍。在从新手到成为 ROS 专业玩家的道路上，理解 ROS 相关的概念是十分必要的。在此基础上，读者能够编写程序来控制实际的机器人或者是模拟的机器人；控制相关的外部设备，如最典型的摄像头。

为了在对相关内容进行介绍时能够更加具体，我们将对 Turtlesim 模拟器进行介绍，并以此为工具结合一系列的例子对 ROS 命令的使用加以介绍。该模拟器是 ROS 组成的一部分，使用者通过模拟器能够很好地了解 ROS 的功能。

本章中，我们将对以下主题进行介绍：

- ROS 是什么、哪些机器人平台支持 ROS。
- 怎样在自己的电脑上安装和启动 ROS。
- ROS 文件目录概览。
- ROS 包、节点和主题介绍。
- 一些常用 ROS 命令的演示示例。
- 怎样在 Turtlesim 模拟器下使用 ROS 命令。

1.1 ROS 的用途以及学习 ROS 的好处

ROS 具备许多操作系统的功能，但是在实际的应用中，它又需要运行在计算机操作系统之上，如 Linux，因此有时会称 ROS 为元操作系统。ROS 的主要用途之一是提供用户、计算机操作系统以及外部设备间的通信功能，这些外部设备包括传感器、摄像机，同时也包括机器人。与其他操作系统一样，ROS 的优点在于其硬件抽象能力，以及它所具备的在不需要用户详尽了解机器人各种细节的前提下对机器人进行控制的能力。

例如,为了移动机器人手臂,可以发出一个 ROS 命令,或者运行机器人设计人员采用 Python 或 C++ 编写的机器人控制脚本。这些脚本还可以调用各种控制机器人手臂做出实际运动的控制程序。而在 ROS 环境下,用户完全可以自行设计和模拟机器人。这些相关主题以及其他一些内容将在本书的后续章节中进行论述。

通过本书,读者能够学习到一系列与 ROS 相关的概念、软件和工具,这些概念、软件和工具正应用在不断增长的各式各样的机器人大军之中。例如,在某一类机器人中应用的导航软件,通过很少的修改,就能够应用在其他机器人上,毕竟空中机器人导航和地面机器人导航是相似的。对于各种各样的机器人而言,各种系统接口正在逐渐的标准化,或者正在升级以应对不断增长的复杂性。对于通用的机器人控制功能,我们能够轻易地找到各种可以使用的程序库。ROS 不仅可以应用在机器人的中心处理中,同时也可以应用在各种类传感器处理和其他子系统的处理中。ROS 的硬件抽象整合了低层次的设备控制与最新技术的快速更新升级。

ROS 是一款开源的机器人软件系统,大学、政府机构以及商业公司等都能够随意使用而不用偿付。开源软件的优势在于用户可以获取系统的源代码,根据自身的实际需要进行修改。更重要的是,对于一些特殊用户,通过引用合适的许可协议,这些软件能够直接应用到商业产品中。用户和公司能够对软件进行改进,添加各种功能模块进去。

在全世界范围内,有成千上万的用户在使用 ROS,同时用户间也在分享使用中的各种经验与知识。这些用户中,既有业余爱好者,同时也有商业机器人的专业开发人员。除了 ROS 研究人员这一庞大的组织之外,还有一个致力于将 ROS 软件应用在机器人制造业之中的 ROS 工业组织。

ROS 的版本控制

ROS 发行版是指一个 ROS 软件包的集合,用户可以将发行版下载到自己的计算机中。这些软件包由 OSRF (Open Source Robotics Foundation, 开源机器人基金会) 提供支持,该组织为一个非盈利组织。ROS 组织定期对发行版进行更新,并对其进行命名。对于 ROS 组织更多的信息,可以查看网址 <http://www.ros.org/about-ros/>。

本书的所有内容都是在 ROS Indigo 下进行编写的,操作系统采用 Ubuntu 14.04。读者请注意检查使用的 Ubuntu 系统或者 ROS 版本的相关更新。

1.2 哪些机器人采用了 ROS

在 ROS 的 wiki 网站 <http://wiki.ros.org/Robots> 上列出了使用 ROS 的机器人型号列表清单。例如,本书中我们使用了四种不同型号的机器人,通过这四种机器人基于 ROS 的控制,为读者提供丰富的 ROS 功能的体验,这四种机器人分别是:

- TurtleBot, 一种移动机器人。