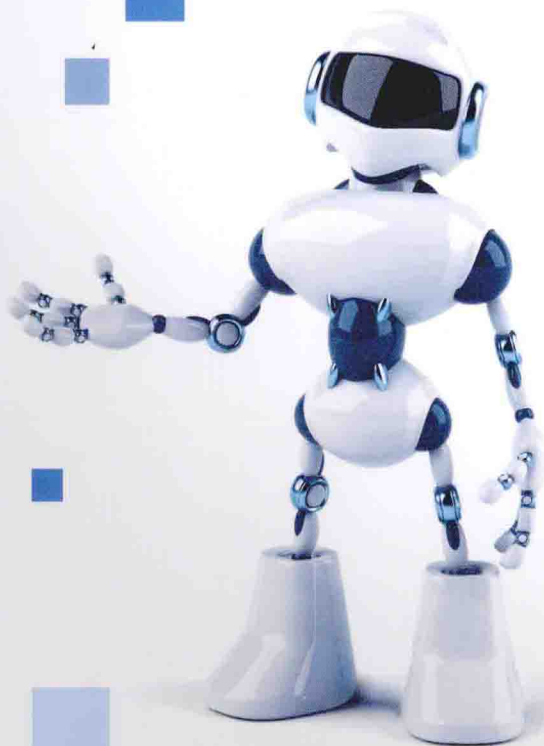


国内首本引进的ROS机器人程序设计译著，让你全面了解 ROS系统的各种工具。

提供了各种实际的示例代码供读者学习和理解ROS的软件框架。

本书可以帮助读者从对ROS一无所知到能够通过ROS系统完成小型机器人系统的开发和编程工作。



ROS机器人程序设计

Learning ROS for Robotics Programming

[西班牙] Aaron Martinez Enrique Fernández 著

刘品杰 译



机械工业出版社
China Machine Press



ROS机器人程序设计

Learning ROS for Robotics Programming

[西班牙] Aaron Martinez Enrique Fern á ndez 著

刘品杰 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

ROS 机器人程序设计 / (西) 马丁内斯 (Martinez, A.) 等著; 刘品杰译. —北京: 机械工业出版社, 2014.8

(电子与嵌入式系统设计丛书)

书名原文: Learning ROS for Robotics Programming

ISBN 978-7-111-47396-1

I. R… II. ① 马… ② 刘… III. 机器人—程序设计 IV. TP242

中国版本图书馆 CIP 数据核字 (2014) 第 161738 号

本书版权登记号: 图字: 01-2014-1838

Aaron Martinez, Enrique Fernández: Learning ROS for Robotics Programming (ISBN: 978-1-78216-144-8)

Copyright © 2013 Packt Publishing. First published in the English language under the title “Learning ROS for Robotics Programming”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2014 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 是一个机器人操作系统框架, 现今已有数百个研究机构和公司将其应用在机器人技术产业中。本书是一本 ROS 的入门手册, 从 ROS 系统的安装到一些常用机器人工具的使用方法如先进的计算机视觉和导航工具等都进行细致的介绍和说明, 并且提供全面的相关示例代码及详细的解释。读者可以按照本书指导进行相关练习。本书适用于机械、自动化、计算机等相关专业高年级本科生及研究生学习, 也适合于准备学习 ROS 的科研人员及企业研发人员学习和使用。

ROS 机器人程序设计

[西班牙] Aaron Martinez 等著

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: 朱 瑛

责任校对: 董纪丽

印 刷: 藁城市京瑞印刷有限公司

版 次: 2014 年 9 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 15

书 号: ISBN 978-7-111-47396-1

定 价: 59.00 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) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@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

The Translator's Words | 译者序

在过去的几十年里，机器人主要是自动化或机械专业的研究领域，计算机往往作为辅助仿真的工具，机器人的程序设计也往往仅限于使用诸如 matlab 机器人工具箱之类的仿真工具。今天很多大型 IT 企业都投入大量资源开发了机器人相关的软件开发工具，例如 Visual Studio 的 Robotics Developer Studio，这些软件工具的目的是颠覆传统的机器人开发和设计模式。机器人操作系统就是基于这个目的产生的。它们能够帮助开发人员在原有机器人的基础上不断地进行二次开发，深入拓展机器人技术。而其中最具代表性的正是开源机器人操作系统 ROS (Robot Operating System)。

与其说 ROS 是一个操作系统，不如说它是一种分布式模块化的开源软件框架。它借用标准的 TCP(UDP)/IP 协议实现了系统内部各个节点之间的通信，网络化的接口允许将第三方组件泛化成为其操作系统的一部分。如果你真正使用过 ROS，就会为其无所不包的开放性而感到惊讶。正因如此，ROS 社区集中了全世界顶尖的机器人研究人员。今天，ROS 能够无缝集成现在已知的大部分机器人操作系统；能满足最高的实时性与可靠性要求；通过使用 EtherCAT 等高速总线能以每秒 Gb 级的速度传输视频数据；能够通过集成其他开源组件及增强现实技术提供多种仿真环境；能够通过多种接口与各种硬件设备或传感器进行通信；能够替你计算繁琐的正逆运动学，通过标定和配置轻松完成视觉伺服等高难度任务；你可以方便地移植其他机器人上已经测试过的各类算法和代码，或引入其他开源库。

虽然 ROS 大幅度降低了开发和应用机器人的难度，但是考虑到机器人领域无所不包，想要学习和使用 ROS 还是有几点需要注意的。首先，你应该学习过 C/C++ 程序设计语言，能够使用 Linux 编译和运行程序，这是阅读本书的基本要求。当然，如果你是一个编程新手，也可以跟着本书示例在实践中学习 Linux 和编程知识。其次，在学习和使用 ROS 的时候，往往还需要懂得程序设计之外的很多机器人知识，尤其是数学知识。但由于机器人领域过于宽泛，因此在学习过程中最好能够专注于某一方面的相关知识，不要贪多求全。再次，ROS 是一个开源软件，因此我们也要保持一种开放的心态。虽然本书能帮助你入门相对轻松一些，但这并不会彻底优化你的学习曲线。如果想要学好 ROS，一定要去 www.ros.org 上学习最

新的资料，尤其是 ROS wiki。最顶尖的 ROS 国际会议是 ROSCON，你可以到 Google 或者 YouTube 上搜索相关内容，里面有 ROS 最新的发展与介绍。最后，本书英文版虽然是 2013 年 9 月出版，但 ROS 和 Linux 一样半年出一个新的版本，因此书中并没有介绍最新的 ROS Hydro 和 ROS Groovy。其实这些并不妨碍你对 ROS 的学习，没有谁为了学习 Linux 编程而非要用最新出版的图书，因为软件的基本框架和组织方式不会发生改变，而且支持 ROS 的机器人硬件并不一定支持最新的 ROS。ROS Hydro 和 ROS Groovy 主要对 ROS 原有的编译系统和部分功能包有所升级和替换，如果有兴趣的话，建议在学习完本书之后尚有余力的情况下，可以通过 wiki 自学 ROS Hydro 或 ROS Groovy。

本书的两位作者并不是以英语为母语的，因此英文原著中存在着较多的语法错误和校对错误。虽然在翻译过程中参考了原著的勘误和所附代码，考虑到译者的水平亦有限，因此仍然不免存在错误，还请大家海涵。在本书之前，ROS 的中文译著甚少，因此有些专业名词的翻译亦有待商榷。为了方便读者学习，译者共享了一个 VMware 9.0 下的虚拟机，里面包含完整安装的 Ubuntu 12.04 和 ROS Fuerte（下载地址是 <http://pan.baidu.com/s/1nt6uH9R>）。这样，使用 VMware 虚拟机的读者可以下载并在 VMware 中运行，然后直接学习第 2 章到第 8 章的内容。关于本书中的各种问题，欢迎大家与我联系，我的邮箱是 liupinjie@gmail.com。

虽然 ROS 是机器人技术，但是其分布式的节点布置同样满足物联网及很多工业应用的需求，同时也有很多外国企业利用它进行机床、船舶、汽车或自动化系统的制造。记住，ROS 代表了未来的智能化时代，赋予了我们新的创造与改变世界的的能力。

刘品杰

2014 年 4 月于北京

Preface | 前言

本书概括性地介绍了 ROS 系统的各种工具。ROS 是一个先进的机器人操作系统框架，现今已有数百个研究团体和公司将其应用在机器人技术产业中。对于机器人技术的非专业人士来说，它也相对容易上手。在本书中，你将了解如何安装 ROS，如何开始使用 ROS 的基本工具，以及如何最终应用先进的计算机视觉和导航工具。

在阅读本书的过程中无需使用任何特殊的设备。书中每一章都附带了一系列的源代码示例和教程，你可以在自己的计算机上运行。这是你唯一需要做的事情。当然，我们还会告诉你如何使用硬件，这样，你可以将你的算法应用到现实环境中。我们在选择设备时特意去选择一些业余用户负担得起的设备，并同时涵盖了在机器人研究中最典型的传感器或执行机构。

最后，由于 ROS 系统的存在使得整个机器人具备在虚拟环境中工作的能力。你将学习如何创建自己的机器人并结合功能强大的导航功能包集。此外如果使用 Gazebo 仿真环境，你将能够在虚拟环境中运行一切。我们会在本书的结尾提供一个能够在 ROS 虚拟环境中进行模拟试验的机器人列表。你将发现你已经可以与机器人一起工作，并理解其背后的原理。

主要内容

第 1 章简单介绍安装 ROS 系统的方法，同时还介绍 ROS 不同版本的安装包的安装，本书使用的是 ROS Fuerte。这一章还会说明如何从 Debian 软件包安装或从源代码进行编译安装，以及在虚拟机中安装。

第 2 章涉及 ROS 框架及相关的概念和工具。该章介绍节点、主题和服务，以及如何使用它们，还将通过一系列示例来说明如何调试一个节点或通过可视化方法直观地查看通过主题所发布的消息。

第 3 章进一步展示 ROS 强大的调试工具，以及通过对节点主题的图形化可以将节点间的通信数据可视化。ROS 提供一个日志记录 API，允许轻松地诊断节点的问题。事实上在使用过程中，我们会看到一些功能强大的图形化工具如 rxconsole 和 rxgraph，以及可视化接口

如 `rxplot` 和 `rviz`。最后介绍如何使用 `rosvbag` 和 `rxrbag` 记录和回放消息。

第 4 章介绍 ROS 系统与真实世界如何连接。这一章介绍在 ROS 下使用的一些常见传感器和执行机构，如激光雷达、伺服电动机、摄像头、RGB-D 传感器等设备。此外，还会解释如何使用嵌入式系统与微控制器，例如非常流行的 `Arduino`。

第 5 章介绍我们在 ROS 系统中实现机器人的第一步是在 ROS 中创建一个机器人模型，包括在 `Gazebo` 仿真环境中如何从头开始对一个机器人进行建模和仿真，并使其在仿真环境中运行。这是后续学习如何使用 ROS 的导航功能包集和其他工具的前提条件。

第 6 章介绍 ROS 对摄像头和计算机视觉任务的支持。首先使用 `FireWire` 和 `USB` 摄像头驱动程序将摄像头连接到计算机并采集图像。然后，你就可以使用 ROS 的标定工具标定你的摄像头。我们会详细介绍和说明什么是图像管道，学习如何使用集成了 `OpenCV` 的多个机器人视觉 API。最后，安装并使用一个视觉测距软件。

第 7 章是本书关于 ROS 导航功能包集的两章中的第 1 章。介绍如何对你的机器人进行使用导航功能包集所需的初始化配置。然后用几个例子对导航功能包集进行说明。

第 8 章延续第 7 章的内容，介绍如何使用导航功能包集使我们的机器人有效地自主导航。本章介绍使用 ROS 的 `Gazebo` 仿真环境和 `rviz` 创建一个虚拟的环境，并在其中构建地图、定位我们的机器人并做路径规划与避障。

第 9 章结合前面几章所学的内容，介绍能够支持 ROS 并使用 `Gazebo` 仿真环境的一些机器人。在该章中，你将看到如何在仿真环境中运行这些机器人，并执行几项本书中介绍过的任务，尤其是与导航功能包集相关的。

预备知识

我们写作本书的目的是希望尽可能让每位读者都可以完成本书的学习并运行示例代码。基本上，你只需要在计算机上安装一个 `Linux` 发行版。虽然每个 `Linux` 发行版本应该都能使用，但还是建议你使用 `Ubuntu` 的最新版。这样你可以根据第 1 章的内容安装 `ROS Fuerte`。对于 ROS 的这一版本，你将需要 `Ubuntu 12.10` 之前的版本，因为之后的版本已经不再支持 `Fuerte` 了。

对于硬件要求，一般来说任何台式计算机或笔记本电脑都满足。但是，最好使用独立显卡来运行 `Gazebo` 仿真环境。此外，如果能够有足够的外围接口将会更好，因为这样你可以连接几个传感器和执行机构，包括摄像头和 `Arduino`。

你还需要 `Git` (`git-core Debian` 软件包)，以便从本书提供的软件源下载源代码。同样，你需要具备 `Bash` 命令行、`GNU/Linux` 工具的基本知识和一些 `C/C++` 编程技巧。

目标读者

本书的目标读者包括所有机器人开发人员，可以是初学者也可以是专业人员。它涵盖了整个机器人系统的各个方面，展示了 ROS 系统如何帮助完成机器人真正自主化的任务。对于听说过 ROS 却从未使用过的机器人领域的学生或科研人员来说，本书将是非常有益的。ROS 初学者能从本书中学习很多 ROS 软件框架的先进理念和工具。不仅如此，经常使用 ROS 的用户也可能从某些章节中学习一些新东西。当然，只有前 3 章是纯粹为初学者准备的，所以那些已经使用过 ROS 的人可以跳过这部分直接阅读后面的章节。

排版说明

在本书中，你会发现一些文本的样式与其他正文内容不同。下面是一些样式的示例及含义。当我们希望能够着重显示示例代码中的某些内容的时候，相关的代码行会如下表示：

```
<launch>
<node pkg="chapter3_tutorials" type="example1"
      name="example1" output="screen"
      launch-prefix="xterm -e gdb --args"/>
</launch>
```

Bash 命令行下输入的命令或显示的结果如下表示：

```
$ rosrun book_tutorials tutorialX _param:=9.0
```



警告或注意事项会这样显示。



提示与小窍门会这样显示。

读者反馈

我们非常欢迎读者能够提出意见与建议。让我们知道你关于本书的一些想法，例如：你喜欢什么，不喜欢什么。读者的反馈对我们是非常重要的，这让我们知道我们最需要做什么。

向我们反馈信息只需发送电子邮件至 feedback@packtpub.com，并在邮件的主题中说明书籍名称。

如果你在某项上有所专长，或者你有兴趣写一本书或者为书籍的出版做出贡献，你可以浏览 www.packtpub.com/authors 阅读我们的作者指南。

读者支持

你现在自豪地拥有了 Packt 出版的书籍，我们会帮助你使你购买的产品物超所值。

源代码下载

可以访问 <http://www.packtpub.com/support>，完成网站注册并通过 E-mail 接收所有代码文件。

彩色图片下载

我们同时提供包含了本书所有彩色的屏幕截图、对话框的 PDF 文件，这些彩色图片能够更好地帮助你理解输出的变化。可以从这里下载这个文件：http://www.packtpub.com/sites/default/files/downloads/1448OS_Graphics.pdf。

勘误

虽然我们已尽力确保本书内容的准确性，但错误在所难免。如果你在我国的书中发现错误，即使是一个错字或一段错误的代码，我们将非常感谢你把错误告知我们。这样可以避免其他读者发生困扰并帮助我们提高本书后续版本的质量。如果你发现任何错误，请访问 <http://www.packtpub.com/submit-errata> 报告这些问题。访问网页，选择你的书，点击勘误提交表格的链接，并输入你的勘误细节。一旦确认了你的勘误，我们会告知你的提交被接受并在我们的网站上更新勘误表。你可以在 <http://www.packtpub.com/support> 根据你选择的标题查看现有的勘误表。

问题

如果关于本书你有任何疑问，请联系我们 questions@packtpub.com，我们会竭尽所能为你解答。

Contents | 目录

译者序

前言

第 1 章 ROS 系统入门 1

1.1 使用软件源安装 ROS Electric 3

1.1.1 添加软件源到 sources.list

文件中 4

1.1.2 设置密码 4

1.1.3 安装 4

1.1.4 环境配置 5

1.2 使用软件源安装 ROS Fuerte 6

1.2.1 配置 Ubuntu 软件源 6

1.2.2 配置 source.list 文件 6

1.2.3 设置密码 7

1.2.4 安装 7

1.2.5 环境配置 8

1.2.6 独立工具 9

1.3 如何安装 VirtualBox 和 Ubuntu 9

1.3.1 下载 VirtualBox 9

1.3.2 创建虚拟机 10

1.4 本章小结 12

第 2 章 ROS 系统架构及示例 13

2.1 理解 ROS 文件系统级 13

2.1.1 功能包 14

2.1.2 功能包集 16

2.1.3 消息类型 16

2.1.4 服务类型 17

2.2 理解 ROS 计算图级 18

2.2.1 节点 19

2.2.2 主题 20

2.2.3 服务 21

2.2.4 消息 22

2.2.5 消息记录包 22

2.2.6 节点管理器 22

2.2.7 参数服务器 22

2.3 理解 ROS 开源社区级 23

2.4 ROS 系统试用练习 23

2.4.1 ROS 文件系统导览 24

2.4.2 创建工作空间 24

2.4.3 创建 ROS 功能包 25

2.4.4 编译 ROS 功能包 26

2.4.5 使用 ROS 节点 26

2.4.6 使用主题与节点交互 28

2.4.7 学习如何使用服务 31

2.4.8 使用参数服务器 33

2.4.9 创建节点 34

2.4.10 编译节点 36

2.4.11	创建 msg 和 srv 文件	37	3.5.2	另一个画图工具 rxtools	60
2.4.12	使用新建的 srv 和 msg 文件	38	3.6	图像可视化	61
2.5	本章小结	42	3.6.1	显示单一图片	61
第 3 章	调试和可视化	43	3.6.2	FireWire 接口摄像头	62
3.1	调试 ROS 节点	44	3.6.3	使用双目立体视觉	63
3.1.1	使用 GDB 调试器调试 ROS 节点	45	3.7	3D 可视化	64
3.1.2	ROS 节点启动时调用 GDB 调试器	46	3.7.1	使用 rviz 在 3D 世界中实现数据可视化	64
3.1.3	设置 ROS 节点 core 文件转存	47	3.7.2	主题与坐标系的关系	66
3.2	调试信息	47	3.7.3	可视化坐标变换	67
3.2.1	输出调试信息	47	3.8	保存与回放数据	68
3.2.2	设置调试信息级别	48	3.8.1	什么是消息记录包文件	69
3.2.3	为特定节点配置调试信息级别	48	3.8.2	使用 rosbag 在包文件中记录数据	69
3.2.4	信息命名	50	3.8.3	回放消息记录文件	70
3.2.5	条件显示信息与过滤信息	50	3.8.4	使用 rxbag 检查消息记录包的主题和消息	71
3.2.6	信息的更多功能——单次显示、可调、组合	51	3.9	rqt 插件与 rx 应用	72
3.2.7	使用 rosconsole 和 rxconsole 在运行时修改调试级别	52	3.10	本章小结	73
3.3	监视系统状态	56	第 4 章	在 ROS 下使用传感器和执行机构	74
3.3.1	节点、主题与服务列表	56	4.1	使用游戏杆或游戏手柄	74
3.3.2	使用 rxgraph 在线监视节点状态图	56	4.1.1	joy_node 如何发送游戏杆动作消息	75
3.4	当奇怪的事情发生——使用 roswtf	58	4.1.2	使用游戏杆数据在 turtlesim 中移动海龟	76
3.5	画标量数据图	58	4.2	使用激光雷达——Hokuyo URG-04lx	79
3.5.1	用 rxplot 画出时间趋势曲线	59	4.2.1	了解激光雷达如何在 ROS 中发送数据	80
			4.2.2	访问和修改激光雷达数据	82

4.3 使用 Kinect 传感器查看 3D 环境	84	5.2.3 加载图形到机器人模型	109
4.3.1 如何发送和查看 Kinect 数据	85	5.2.4 使机器人模型运动	109
4.3.2 创建和使用 Kinect 示例	86	5.2.5 物理和碰撞属性	110
4.4 使用伺服电动机——Dynamixel	88	5.3 xacro——一个写机器人模型的更好方法	111
4.4.1 Dynamixel 如何发送和接收运动命令	89	5.3.1 使用常量	111
4.4.2 创建和使用伺服电动机示例	90	5.3.2 使用数学方法	112
4.5 使用 Arduino 添加更多的传感器和执行机构	91	5.3.3 使用宏	112
4.6 使用惯性测量模组——Xsens MTi	94	5.3.4 使用代码移动机器人	112
4.6.1 Xsens 如何在 ROS 中发送数据	95	5.3.5 使用 SketchUp 进行 3D 建模	116
4.6.2 创建和使用 Xsens 示例	96	5.4 在 ROS 中仿真	117
4.7 使用低成本惯性测量模组 IMU-10 自由度	98	5.4.1 在 Gazebo 中使用 URDF3D 模型	117
4.7.1 下载加速度传感器库	99	5.4.2 在 Gazebo 中添加传感器	120
4.7.2 Arduino Nano 和 10 自由度传感器编程	99	5.4.3 在 Gazebo 中加载和使用地图	121
4.7.3 创建 ROS 节点并使用 10 自由度传感器数据	101	5.4.4 在 Gazebo 中移动机器人	123
4.8 本章小结	103	5.5 本章小结	125
第 5 章 3D 建模与仿真	104	第 6 章 机器视觉	126
5.1 自定义机器人在 ROS 中的 3D 模型	104	6.1 连接和运行摄像头	128
5.2 创建第一个 URDF 文件	104	6.1.1 FireWire IEEE1394 摄像头	128
5.2.1 解释文件格式	106	6.1.2 USB 摄像头	132
5.2.2 在 rviz 里查看 3D 模型	107	6.2 使用 OpenCV 制作 USB 摄像头驱动程序	133
		6.2.1 创建 USB 摄像头驱动功能包	134
		6.2.2 使用 ImageTransport API 发布摄像头帧	135
		6.2.3 使用 cv_bridge 进行 OpenCV 和 ROS 图像处理	138

6.2.4	使用 ImageTransport 发布 图像	139	7.6.1	使用 map_server 保存 地图	181
6.2.5	在 ROS 中使用 OpenCV	139	7.6.2	使用 map_server 加载 地图	182
6.2.6	显示摄像头输入的图像	140	7.7	本章小结	183
6.3	如何标定摄像头	140	第 8 章 导航功能包集进阶		184
6.4	ROS 图像管道	147	8.1	创建功能包	184
6.5	对于计算机视觉任务有用的 ROS 功能包	152	8.2	创建机器人配置	184
6.6	使用 viso2 执行视觉测距	153	8.3	配置全局和局部代价地图	187
6.6.1	摄像头位姿标定	154	8.3.1	基本参数的配置	187
6.6.2	运行 viso2 在线演示	156	8.3.2	全局代价地图的配置	188
6.6.3	使用低成本双目摄像头 运行 viso2	158	8.3.3	局部代价地图的配置	189
6.7	本章小结	159	8.4	基本局部规划器配置	189
第 7 章 导航功能包集入门		160	8.5	为导航功能包集创建启动文件	190
7.1	ROS 导航功能包集	160	8.6	为导航功能包集设置 rviz	191
7.2	创建转换	161	8.6.1	2D 位姿估计	191
7.2.1	创建广播机构	162	8.6.2	2D 导航目标	192
7.2.2	创建侦听器	162	8.6.3	静态地图	193
7.2.3	查看坐标变换树	164	8.6.4	点云	193
7.3	发布传感器信息	165	8.6.5	机器人立足点	193
7.4	发布里程数据	168	8.6.6	障碍	194
7.4.1	Gazebo 如何获取里程 数据	169	8.6.7	膨胀障碍	194
7.4.2	创建自定义里程数据	171	8.6.8	全局规划	195
7.5	创建基础控制器	175	8.6.9	局部规划	195
7.5.1	使用 Gazebo 创建里程 数据	176	8.6.10	规划器规划	196
7.5.2	创建基础控制器	178	8.6.11	当前目标	196
7.6	使用 ROS 创建地图	180	8.7	自适应蒙特卡罗定位	197
			8.8	避免障碍	199
			8.9	发送目标	200
			8.10	本章小结	202

第 9 章 在实践中学习	203	型人形机器人	217
9.1 REEM——类人形 PAL 机器人	204	9.3.1 从软件源安装 Robonaut 2	217
9.1.1 从官方软件源安装 REEM	205	9.3.2 在国际空间站的固定支座 上运行 Robonaut2	218
9.1.2 使用 Gazebo 仿真环境 运行 REEM	208	9.4 Husky——Clearpath 的轮式 机器人	222
9.2 PR2——柳树车库机器人	210	9.4.1 安装 Husky 仿真环境	222
9.2.1 安装 PR2 仿真环境	210	9.4.2 运行 Husky 仿真环境	222
9.2.2 在仿真环境中运行 PR2	211	9.5 TurtleBot——低成本移动 机器人	224
9.2.3 生成地图与定位	214	9.5.1 安装 TurtleBot 仿真环境	224
9.2.4 在仿真环境中运行 PR2 演示程序	216	9.5.2 运行 TurtleBot 仿真环境	224
9.3 Robonaut 2——NASA 的敏捷		9.6 本章小结	225

第 1 章

ROS 系统入门

欢迎来到本书的第 1 章。本章会介绍如何安装 ROS 系统。与其说 ROS 是一个操作系统，不如说它是一种新的标准化机器人软件框架。通过 ROS，你可以使用大量的示例代码和开源程序轻松地完成机器人编程和控制。同时，你还能够理解如何使用各种传感器与执行机构，并为你的机器人增加如自动导航和视觉定位等新的功能。在这里，我们要感谢相关的基金会及社区对开源软件的贡献，并不断地开发出最新的算法和功能使 ROS 系统不断进步。

本书包含以下知识：

- 在特定版本的 Ubuntu 系统下安装 ROS 框架
- 学习 ROS 的基本操作
- 调试数据和数据可视化
- 在 ROS 框架下对机器人编程
- 创造 3D 模型并进行仿真
- 使用导航功能包集使机器人进行自动导航

在本章中，首先要在 Ubuntu 系统中安装完整版本的 ROS。Ubuntu 不仅能够全面支持 ROS，而且它也是 ROS 官方推荐的操作系统。当然，你也可以在其他的操作系统中安装 ROS。但是在其他操作系统中，ROS 大多是实验性质的移植版本，因此容易出现各种安装和程序执行错误。因此在学习和使用本书的过程中，推荐使用 Ubuntu。

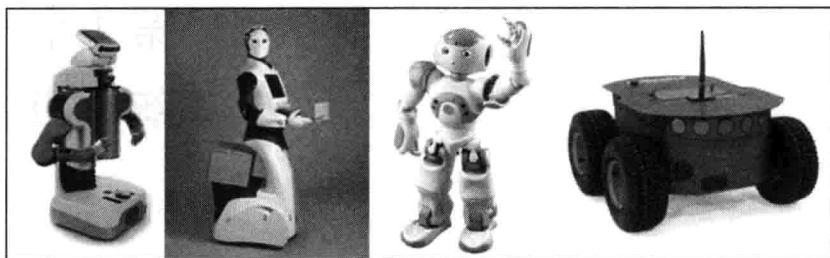
在开始安装之前，首先了解一下 ROS 的历史。

Robot Operating System (ROS) 是一种得到广泛使用的机器人操作与控制系统软件框架。该框架使用了当前最流行的面向服务 (SOA) 的软件技术，通过网络协议将节点间数据通信解耦。这样就能够轻松地集成不同语言不同功能的代码。ROS 的基本原理是无需改动就能够在不同的机器人上复用代码。基于这些，我们就可以在不同的机器人上分享和复用已经实现的功能，而不需要做太多的工作，避免了重复劳动。

2007 年，斯坦福大学人工智能实验室 (SAIL) 在斯坦福 AI 机器人项目 (Stanford AI Robot project) 的支持下开发了 ROS。2008 年之后，则主要在 Willow Garage 公司支持下与其他 20 多家研究机构进行联合研发 ROS。

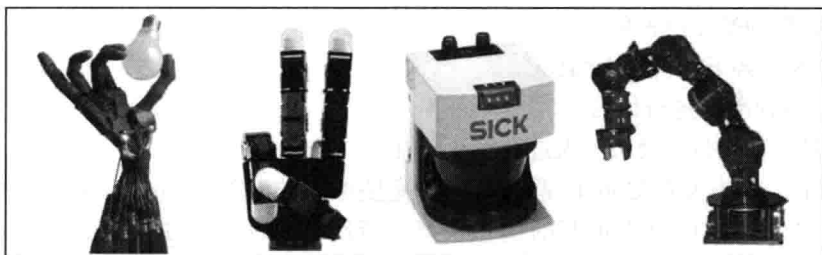
现在已经有非常多研究机构通过增加 ROS 支持的硬件或开放软件源代码的方式加入 ROS 系统的开发中。同样，也有很多家公司将其产品逐步进行软件迁移并在 ROS 系统中应

用。部分 ROS 系统支持的平台如下图所示。这些平台往往会开放大量的代码、示例和仿真环境，以便开发人员轻松地开展工作。

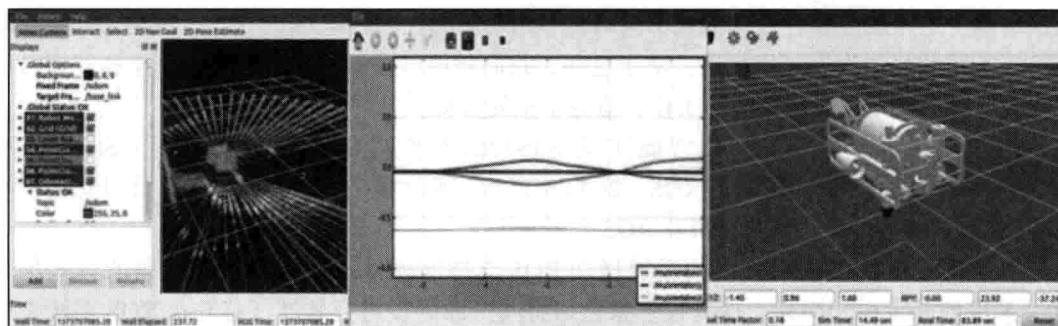


ROS 系统已经支持这些机器人中的传感器和执行机构，同时每天 ROS 软件框架所支持的设备也在大量增加。

ROS 提供了一个标准的操作系统环境，包括硬件抽象、底层设备控制、通用功能的实现、进程间消息转发和功能包管理等。它将多路复用传感器、控制器、运动状态、规划目标、执行机构及其他设备全部抽象成节点 (node)，并能对各个节点进程间消息的发生和接受的处理过程通过节点状态图展示。它的各种库都是面向类 Unix 系统的 (ROS 主要支持 Ubuntu Linux，而其他系统如 Fedora 和 Mac OS X 则是实验性质的)。



*-ros-pkg 包主要作为开发高级库的基础软件源。其很多功能是和 ROS 系统绑定的，如导航库和 rviz 可视化界面都基于这个源。其中的一些库包含很多强大的工具，可以帮助我们方便使用 ROS 并了解机器人的实时状态。其中，可视化工具、仿真环境和调试工具是最重要的几个。



ROS 是一个使用 BSD (Berkeley Software Distribution) 开源协议的开源软件。无论是商业应用，还是科学研究，它都是免费的。*-ros-pkg 包受到了多个开源协议的限制。

ROS 致力于机器人的代码复用，这样每次研发新的机器人时，开发人员和科学家们再也不必全部推倒重来。通过 ROS，你可以集中精力做更多的事情，应用不同代码库中的代码，再完善，并再次分享。

ROS 已经发布了多个版本，最新的版本是 Groovy。在本书中，我们使用的版本是 Fuerte，因为这个版本更加稳定。因此需要说明的是，本书中的示例和教程不一定都能够在 Groovy 下使用。

下面我们会介绍如何安装 Electric 版本和 Fuerte 版本的 ROS。不同的 ROS 版本对应不同的 Ubuntu 版本，所以某些机器人可能仍然使用老版本的 Ubuntu。即使在本书中我们使用 Fuerte，但是在实际工作中，你仍然可能需要安装 Electric 以便运行一些老版本的代码。

如前文所述，本书中所使用的操作系统是 Ubuntu，如果你习惯使用其他操作系统又想完成本书的学习，最好的选择就是安装一个带有 Ubuntu 的虚拟机。因此，我会在最后介绍虚拟机的安装方法。

当然，如果你想在其他系统中安装 ROS，你可以根据链接 <http://wiki.ros.org/fuerte/Installation> 中的指导来完成。

1.1 使用软件源安装 ROS Electric

安装 ROS 有多种方法，你可以按照本书的方法直接使用软件源，也可以使用源文件并编译后安装。使用软件源的方法相对安全，不容易出现错误。

按照本节介绍的步骤，你可以在自己的电脑上逐步安装 ROS Electric，安装过程详见官网页面 <http://wiki.ros.org/electric/Installation>。

我们假设你知道 Ubuntu 软件源 (repository) 的含义，并且知道如何管理它。如果你有任何疑问请查询 <https://help.ubuntu.com/community/Repositories/Ubuntu> 了解相关知识。

在开始安装之前，需要首先配置软件源，为此需要先把软件源属性设为 restricted、universal、multiverse，为了检查你的 Ubuntu 版本是否支持这些软件源，请双击打开你桌面左面的 Ubuntu 软件中心 (Ubuntu Software Center)，如右图所示。

打开 Edit | Software Sources 标签页，你能看到以下界面，你要保证各个选项与下图中一致。

