

机器人操作系统浅析

> A Gentle Introduction to ROS

[美] 奥凯恩 M. 杰森(Jason M. O'Kane) 著

肖军浩 译



国防工业出版社

National Defense Industry Press

机器人操作系统浅析

[美] 奥凯恩 M. 杰森 (Jason M. O'Kane) 著

肖军浩 译

国防工业出版社

· 北京 ·

著作权合同登记 图字:军-2016-050 号

图书在版编目(CIP)数据

机器人操作系统浅析/(美)杰森 M. 奥凯恩(Jason M. O' Kane)著;肖军浩译. —北京:国防工业出版社,2016.9

书名原文:A Gentle Introduction to ROS

ISBN 978-7-118-11056-2

I. ①机… II. ①杰… ②肖… III. ①机器人-操作系统
IV. ①TP242

中国版本图书馆 CIP 数据核字(2016)第 212644 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京嘉恒彩色印刷有限责任公司

新华书店经售

*

开本 710×1000 1/16 印张 10 字数 180 千字

2016 年 9 月第 1 版第 1 次印刷 印数 1—1500 册 定价 39.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

前 言

因为集成了全世界机器人领域顶级科研机构,包括斯坦福大学、麻省理工学院、慕尼黑工业大学、加州大学伯克利分校、佐治亚理工大学、弗莱堡大学、东京大学等多年的研究成果,开源机器人操作系统(Robot Operating System, ROS)甫一问世便受到了科研人员的广泛关注。随后,ROS 又借助开源的魅力吸引了世界各地机器人领域的仁人志士群策群力,推动其不断进步。2013 年麻省理工学院科技评论(MIT Technology Review)指出:“从 2010 年发布 1.0 版本以来,ROS 已经成为机器人软件的事实标准(*de facto* standard)”。

译者在德国汉堡大学攻读博士学位期间,有幸于 2010 年成为 ROS 的第一批用户,并将其介绍给身边的同事以及国防科技大学的老师 and 同学。2013 年,译者所在研究团队将 NuBot 中型组足球机器人的软件系统移植到了 ROS 框架下,并于 2014 年和 2015 年分别参加了在巴西若昂佩索阿和中国合肥举办的机器人足球世界杯。使用 ROS 后 NuBot 足球机器人软件系统的鲁棒性、易用性和可维护性均有大幅度提高。对我们将 ROS 用于中型组足球机器人的工作,国际同行给予了非常积极的评价。

译者连续两年将 ROS 的使用作为国防科技大学高年级本科生实践教学的重要环节,发现一个普遍问题:学生能够迅速理解 ROS 的框架结构和基本概念,但是实际使用时问题却层出不穷,而其中大部分是共性问题。O'Kane 教授这本“A Gentle Introduction to ROS”对 ROS 初学过程中的常见问题做了全面的总结。当然,这本书不仅仅是常见问题汇编,它还对概念和工具做了比在线文档更深入的剖析。故而,译者在征求 O'Kane 教授的同意后,将该书译为中文,供中国的 ROS 初学者在实际使用时参考。

首先,感谢 O'Kane 教授对于本书翻译工作的肯定和支持。其次,本书的翻译得到国防科学技术大学“控制科学与工程高级专题”课程师生的大力支持,其中赵云云、李峻翔、肖志鹏、贾凡、朱琪、郭昭宇、王志强、陈春玉、魏

翔宇分别参与了部分章节的翻译工作。此外,王祥科副教授审阅了初稿并提出了许多宝贵的意见,对此,译者表示诚挚的谢意。感谢 NuBot 研究小组 (<http://nubot.trustie.net>) 全体成员对于本书的支持和帮助,尤其要感谢的是卢惠民副教授以及博士研究生曾志文和硕士研究生黄玉玺。最后,感谢家人对我工作的关心、理解和支持。特别感谢我的夫人王芳,是她一直对我的工作给予了全力的支持,主动承担起了照顾家庭和女儿的责任。还有可爱的女儿肖艺涵,她的成长给我带来了无尽的快乐和惊喜。

限于译者水平,书中难免会有不足之处,热切地希望得到各位读者的宝贵意见。作者的 E-mail 地址是: junhao.xiao@ieee.org。

译者

2015 年 9 月于长沙

目 录

第 1 章 绪论	1
1.1 选择 ROS 的理由	1
1.2 内容概述	4
1.3 行文约定	6
1.4 更多信息	6
1.5 下一章简介	8
第 2 章 入门概述	9
2.1 安装 ROS	9
2.2 配置账户	12
2.3 使用 turtlesim 的小例子	13
2.4 功能包/软件包(Packages)	14
2.5 节点管理器(The Master)	18
2.6 节点(Nodes)	19
2.7 话题和消息	22
2.7.1 查看节点构成的计算图	22
2.7.2 消息和消息类型	25
2.8 一个更复杂的例子	32
2.8.1 话题通信的多对多机制	34
2.8.2 节点之间的松耦合关系	34
2.9 问题检查	35
2.10 展望	36
第 3 章 编写 ROS 程序	37
3.1 创建工作区和功能包	37
3.2 你好,ROS!	39

3.2.1	一个简单的程序	39
3.2.2	编译 Hello 程序	42
3.2.3	执行 hello 程序	46
3.3	发布者程序	46
3.3.1	发布消息	48
3.3.2	消息发布循环	51
3.3.3	编译 pubvel	53
3.3.4	执行 pubvel	54
3.4	订阅者程序	55
3.4.1	编译并运行 subpose	60
3.5	展望	60
第 4 章	日志消息	61
4.1	严重级别	61
4.2	示例程序	62
4.3	生成日志消息	64
4.4	查看日志消息	69
4.4.1	控制台	69
4.4.2	rosout 上的消息	70
4.4.3	日志文件	73
4.5	启用和禁用日志消息	74
4.6	展望	78
第 5 章	计算图源命名	79
5.1	全局名称	79
5.2	相对名称	81
5.3	私有名称	83
5.4	匿名名称(Anonymous names)	84
5.5	展望	85
第 6 章	启动文件	86
6.1	使用启动文件	86
6.2	创建启动文件	89
6.2.1	启动文件的存储位置	89
6.2.2	启动文件的基本元素	90

6.3	在命名空间内启动节点	94
6.4	名称重映射(Remapping names)	97
6.4.1	创建重映射	98
6.4.2	反向海龟(Reversing a turtle)	99
6.5	启动文件的其他元素	103
6.5.1	包含其他文件	103
6.5.2	启动参数(Launch arguments)	105
6.5.3	创建组(Creating groups)	107
6.6	展望	109
第7章	参数	110
7.1	通过命令行获取参数	110
7.2	例: turtlesim 中的参数	113
7.3	使用 C++ 获取参数	115
7.4	在启动文件中设置参数	119
7.5	展望	122
第8章	服务	123
8.1	服务的专用术语	124
8.2	从命令行查看和调用服务	124
8.3	客户端程序	129
8.4	服务器程序	135
8.4.1	运行和改进服务器程序	138
8.5	展望	140
第9章	消息录制与回放	141
9.1	录制与回放包文件	141
9.2	示例: 正方形运动轨迹的包文件	143
9.3	启动文件里面的包文件	147
9.4	展望	148
第10章	总结	149
10.1	下一步	149
10.2	展望	152

第 1 章 绪 论[†]

合抱之木，生于毫末；九层之台，起于累土；千里之行，始于足下。

——老子

本章主要介绍 ROS 系统的优势和本书的框架结构。

1.1 选择 ROS 的理由

近年来，机器人领域取得了举世瞩目的进展。性价比较高的机器人平台，包括地面移动机器人、旋翼无人机和类人机器人等，得到了广泛应用。更令人感到振奋的是，越来越多的高级智能算法让机器人的自主等级逐步提高。

尽管如此，对于机器人软件开发人员来说，仍然存在着诸多挑战。本书主要介绍一个软件平台，即机器人操作系统¹(Robot Operating System, ROS)，它可以帮助提高机器人软件的开发效率。ROS 系统的官方定义如下：

ROS 是面向机器人的开源的元操作系统(meta-operating system)¹。它能够提供类似传统操作系统的诸多功能，如硬件抽象、底层设备控制、常用功能实现、进程间消息传递和软件包管理等。此外，它还提供相关工具和库，用于获取、编译、编辑代码以及在多台计算机之间运行程序完成分布式计算。

虽然上述定义很准确，强调了 ROS 与传统操作系统的异同，但可能仍然无法让读者抓住 ROS 的核心要义，尤其是使用 ROS 能够给机器人软件开发带来哪些优势。一般而言，学习一个新的系统框架，特别是 ROS 这样复杂多样的框

[†]本章由肖军浩、赵云云翻译。

¹ <http://wiki.ros.org/ROS/Introduction>

架，往往要耗费大量的时间和精力，因此我们必须确保付出这个代价是有意义的。下面简单列举几个使用 ROS 能够解决的机器人软件开发问题。

分布式计算 现代机器人系统往往需要多台计算机同时运行多个进程，例如：

- ☞ 一些机器人搭载多台计算机，每台计算机用于控制机器人的部分驱动器或传感器；
- ☞ 即使只有一台计算机，通常仍将程序划分为独立运行且相互协作的小的模块来完成复杂的控制任务，这也是常见的做法；
- ☞ 当多个机器人需要协同完成一个任务时，往往需要互相通信来支撑任务的完成；
- ☞ 用户通常通过台式机、便携式计算机或者移动设备发送指令控制机器人，这种人机交互接口可以认为是机器人软件的一部分。

单计算机或者多计算机不同进程间的通信问题是上述例子中的主要挑战。ROS 为实现上述通信提供了两种相对简单、完备的机制，我们将在第 3 章和第 8 章进行详细讨论。

软件复用 随着机器人研究的快速推进，诞生了一批应对导航、路径规划、建图等通用任务的算法。当然，任何一个算法实用的前提是其能够应用于新的领域，且不必重复实现。事实上，如何将现有算法快速移植到不同系统一直是一个挑战，ROS 通过以下两种方法解决这个问题。

- ☞ ROS 标准包(Standard Packages)提供稳定、可调式的各类重要机器人算法实现。
- ☞ ROS 通信接口正在成为机器人软件互操作的 **事实标准**，也就是说绝大部分最新的硬件驱动和最前沿的算法实现都可以在 ROS 中找到。例如，在 ROS 的官方网页¹上有着大量的开源软件库，这些软件使用 ROS 通用接口，从

¹ <http://wiki.ros.org/browse>

而避免为了集成它们而重新开发新的接口程序。

综上所述,开发人员如果使用 ROS 可以——当然,在具备 ROS 基础知识后——将更多的时间用于新思想和新算法的设计与实现,尽量避免重复实现已有的研究结果。

快速测试 为机器人开发软件比其他软件开发更具挑战性,主要是因为调试准备时间长,且调试过程复杂。况且,因为硬件维修、经费有限等因素,不一定随时有机器人可供使用。ROS 提供两种策略来解决上述问题。

精心设计的 ROS 系统框架将底层硬件控制模块和顶层数据处理与决策模块分离,从而可以使用模拟器替代底层硬件模块,独立测试顶层部分,提高测试效率。

ROS 另外提供了一种简单的方法可以在调试过程中记录传感器数据及其他类型的消息数据,并在试验后按时间戳回放。通过这种方式,每次运行机器人可以获得更多的测试机会。例如,可以记录传感器的数据,并通过多次回放测试不同的数据处理算法。在 ROS 术语中,这类记录的数据叫作包(bag),一个被称为 rosbag 的工具可以用于记录和回放包数据,见第 9 章。

采用上述方案的一个最大优势是实现代码的“无缝连接”,因为实体机器人、仿真器和回放的包可以提供同样(至少是非常类似)的接口,上层软件不需要修改就可以与它们进行交互,实际上甚至不需要知道操作的对象是不是实体机器人。

当然,ROS 操作系统并不是唯一具备上述能力的机器人软件平台。在笔者看来,ROS 的最大不同在于来自机器人领域诸多开发人员的认可和支持,这种支持将促使 ROS 的未来不断发展、完善、进步。

对 ROS 的误解…… 最后,让我们用一点时间来澄清容易对 ROS 产生的误解,从另一个侧面来认识 ROS。

- ☞ ROS 不是一种编程语言。实际上，ROS 的主要代码由 C++语言¹编写，本书后续章节也会介绍如何在 ROS 中使用 C++进行编程。客户端库的编写还可以使用 Python²、Java³和 Lisp⁴等其他多种语言⁵编写。
- ☞ ROS 不仅是一个函数库，除包含客户端库(Client Libraries)外，还包含一个中心服务器(Central Server)、一系列命令行工具、图形化界面工具以及编译环境。
- ☞ ROS 不是集成开发环境。虽然 ROS 没有规定软件开发环境，但几乎所有的主流 IDE⁶都可用于基于 ROS 的软件开发。此外，我们还可以根据个人喜好，使用普通的文本编辑器和命令行来完成相应的开发，而无需任何 IDE。

1.2 内容概述

本书的主要目的是对 ROS 的概念和技术进行整体介绍，从而使读者具备独立编写 ROS 软件的基本技能，但并不涵盖以下几点：

- ☞ 本书不是一本介绍如何编程的书籍。我们打算详细讨论编程的基本概念。因此，本书假设读者已经具备阅读、编写和理解 C++代码的基本能力。
- ☞ 本书不是一本参考手册。关于 ROS 已有大量的资料可以查阅，包括在线指导手册⁷和参考资料⁸。本书无意也不可能取代这些资源。相反，本书只是介绍 ROS 最基本的特征，为后面更好地使用 ROS 提供一个好的开端。
- ☞ 本书不是一本关于机器人算法的书籍。机器人学的研究，尤其是自主机器人控制算法的研究，可以说是相当有趣的，且针对这些问题已经提出了令

1 <http://wiki.ros.org/roscpp>

2 <http://wiki.ros.org/rospy>

3 <http://wiki.ros.org/rosjava>

4 <http://wiki.ros.org/roslisp>

5 <http://wiki.ros.org/ClientLibraries>

6 <http://wiki.ros.org/IDEs>

7 <http://wiki.ros.org/ROS/Tutorials>

8 <http://wiki.ros.org/APIs>

人眼花缭乱的算法。本书并不针对某个具体算法进行讲解，而是告诉读者如何使用好 ROS 这一工具，更好地实现和测试感兴趣的算法。

章节及其依赖关系 本书的基本构架如图 1.1 所示，其中章节用矩形框表示，箭头描述它们之间的主要依赖关系，读者可以在这些约束下按照不同顺序阅读本书。

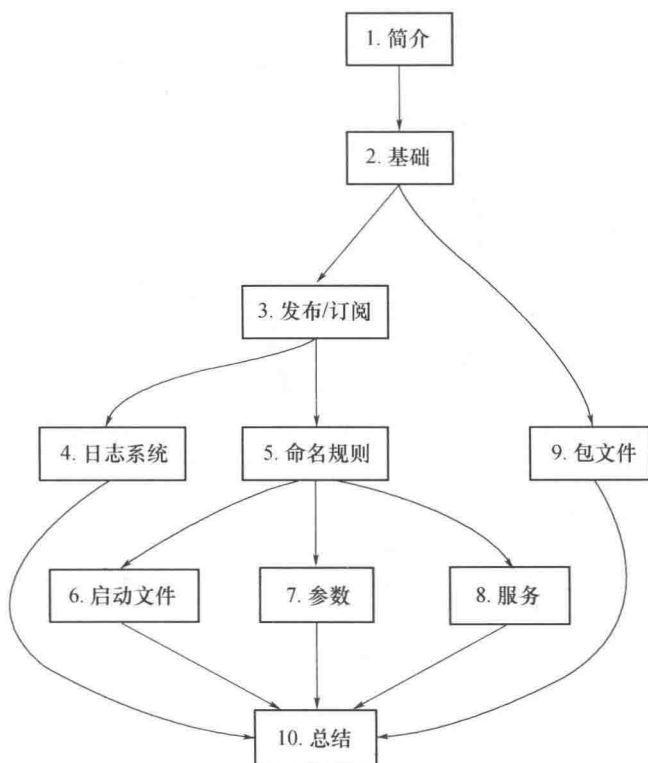


图 1.1 章节之间的相互依赖关系

目标读者 本书能够为从事机器人技术相关领域的学生、研究人员和爱好者提供一个快速上手 ROS 的参考手册。我们希望读者对 Linux 系统有所了解(如基本的命令行操作、安装软件、编辑文件和设置环境变量等)，熟悉 C++，并且对通过编程控制机器人感兴趣。我们进一步假设读者使用的 Linux 操作系统版本是 Ubuntu 14.04(在成文过程中最新的官方支持版本)，并且命令行使用 bash shell。然而，这些选择对其他系统配置并不会造成太大影响，使用其他 Linux 发行版(尤其是基于 deb 包的发行版)和 shell 通常不会有问题。

1.3 行文约定

在全书中，我们尝试预先提示一些可能出现的共性问题。这类警告是值得关注的，特别是当运行结果与期望有出入时，使用如下符号标记：



这个危险符号提示读者可能会出现错误。

此外，某些章节中包含了只有部分读者可能会感兴趣的内容，这些内容对于概念理解没有影响，用如下符号进行标记：



该快进符号用于描述一些阅读过程中，尤其是初次接触本书，暂时可以跳过的内容。

1.4 更多信息

如前所述，本书并不是一本完备的 ROS 参考手册。几乎可以肯定，要真正动手使用 ROS 将要了解更多的细节。幸运的是，互联网上有关 ROS 的信息十分丰富。

☞ 最为重要的是 ROS 开发联盟维护的维基文档¹，其中包括大量教程。本书给出了其中一些页面的链接，在书中用脚注符号标记。如果使用主流的 PDF 阅读器阅读本书的电子版，则可以直接点击该符号在浏览器中打开对应网页。

☞ 如果在调试过程中遇到问题，可以登录类似 Stack Exchange(译者注：Stack Exchange 是一系列的问答网站，每一个网站包含不同领域的问题)风格的 Q&A 网站²进行查询和提问。

¹ <http://wiki.ros.org>

² <http://answers.ros.org>

☞ 另一个有效获取最新信息的途径就是 ROS 用户(ros-users)邮件列表¹。
这里主要介绍两个细节以便读者明确相关概念。

发行版本(Distributions) ROS 的主要版本称为发行版,其版本号²以顺序字母作为版本名的首字母来命名(这种方式与其他大型工程的版本命名方式类似,如 Ubuntu、Android)。在书写本书的过程中,当前版本号 indigo,其后续版本被命名为 jade,将于 2015 年 5 月发布³(译者注:事实上,ROS jade 已经于 2015 年 5 月 23 日按计划日期发布)。ROS 的旧版本号包括 hydro、groovy、feuerte、electric、diamondback、C Turtle 和 box turtle。这些名称将会多次出现在本书中。

🔍 为了确保简单和时效性,本书假设读者使用 ROS indigo。

▶▶ 如果需要在某些情况下使用 hydro 版本,本书中的绝大多数例子可以不经修改就可以使用。上述结论对 groovy 版本同样奏效,但有一点需要特别注意:在 groovy 之后的版本中,对于 turtlesim 仿真器的速度指令已经更新为用于很多实体机器人的标准信息类型和话题名称。其区别如下表所示:

版本	话题名称	消息类型
groovy	/turtle1/command_velocity	Turtlesim/Velocity
indigo, hydro	/turtle1/cmd_vel	geometry_msgs/Twist

这个改变将引起一些实现上的变化:

☞ 在第 40 页,当添加依赖库时,需要使用 turtlesim 来替换 geometry_msgs。

☞ 在第 49 页,相应的头文件也要替换为

turtlesim/Velocity.h

而不再使用

1 <http://lists.ros.org/mailman/listinfo/ros-users>

2 <http://wiki.ros.org/Distributions>

3 <http://wiki.ros.org/indigo/Planning>

geometry_msgs/Twist.h

☞ turtlesim/Velocity 消息类型只包含两个域，即 `linear` 和 `angular`。上述两个域与 `geometry_msgs/Twist` 中的 `linear.x` 和 `angular.z` 含义相同。这个改变同样出现在第 28 页的命令行和第 47 页的 C++ 代码中。

编译系统(Build Systems) 从 `groovy` 版本开始，ROS 对软件的编译进行了重大改动。在 `groovy` 及之前的版本中，ROS 采用 `rosbuild` 系统来完成软件的编译，而在新的版本中，则改用 `catkin` 编译系统。了解这一点非常重要，尤其在阅读参考教程时需要注意，编译系统的改变导致很多教程分为 `rosbuild` 和 `catkin` 两个版本。版本的选择可以通过操作教程页面顶端的按钮完成。虽然在有些情况下，`rosbuild` 是更好的选择¹，但本书主要介绍 `catkin` 系统的使用。

1.5 下一章简介

在下一章，我们学习 ROS 的基本概念和工具，并开始使用 ROS。

¹ http://wiki.ros.org/catkin_or_rosbuild

第2章 入门概述[†]

工欲善其事，必先利其器。

——孔子

本章中，我们将简述 ROS 安装过程，介绍一些 ROS 的基本概念，并与运行中的 ROS 系统进行一些基本的交互。

在介绍用 ROS 写程序的细节之前，我们有必要先了解如何启动、运行 ROS，并理解 ROS 使用的一些基本概念。作为后续内容的基础，本章首先简单介绍如何安装 ROS 和配置用户账户，接着我们将浏览一个运行中的 ROS 系统（更确切地说，是一个 turtlesim 仿真器），最后将学习如何用命令行工具与这个系统进行交互。

2.1 安装 ROS

当然，使用 ROS 之前，必须首先确认已经在计算机上成功安装了 ROS 软件。如果你正在使用一台别人已经安装过 ROS 的计算机，包括我们后续章节将频繁使用的 `ros-indigo-turtlesim` 功能包，那么你可以直接跳到 2.2 节。事实上，安装过程有详细的帮助文档，而且大部分都很简单^{1, 2}。下面是对一些必要步骤的小结。

添加 ROS 软件源 使用超级用户(译者注：Ubuntu 下使用 `sudo` 命令得到超级用户权限)在根目录下，创建文件

[†]本章由肖军浩、李峻翔、肖志鹏翻译。

¹ <http://wiki.ros.org/ROS/Installation>

² <http://wiki.ros.org/indigo/Installation/Ubuntu>