

ROS 入门与实战

李振伟 陈萌 马庆华 于泽宽 编著



中国矿业大学出版社

ROS 入门与实战

李振伟 陈 萌 马庆华 于泽宽 编著

中国矿业大学出版社

内 容 简 介

本书深入浅出地介绍了ROS的特点、基本使用方法以及实际应用,内容涉及移动机器人自主导航、视觉识别、机械臂运动规划等,可帮助不同领域的机器人开发人员了解并熟练使用ROS。

本书内容分为6章。其中第1章主要介绍ROS的基本概念、安装方法和常用命令。第2章主要介绍ROS 2.0的基本使用方法。第3章主要介绍ROS中的仿真工具。第4章主要介绍抓取仿真和操作,第5章介绍利用ROS实现 Turtlebot 移动机器人的自主导航与避障。第6章主要介绍采用 MoveIt! 强大的功能实现多自由度机械臂的运动规划。

图书在版编目(CIP)数据

ROS入门与实战 / 李振伟等编著. —徐州:中国矿业大学出版社,2016.4

ISBN 978 - 7 - 5646 - 3040 - 9

I. ①R… II. ①李… III. ①机器人—程序设计
IV. ①TP242

中国版本图书馆CIP数据核字(2016)第035374号

书 名 ROS入门与实战
编 著 李振伟 陈 萌 马庆华 于泽宽
责任编辑 杨 洋
出版发行 中国矿业大学出版社有限责任公司
(江苏省徐州市解放南路 邮编 221008)
营销热线 (0516)83885307 83884995
出版服务 (0516)83885767 83884920
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 徐州中矿大印发科技有限公司
开 本 787×960 1/16 印张 9.25 字数 166千字
版次印次 2016年4月第1版 2016年4月第1次印刷
定 价 18.00元

(图书出现印装质量问题,本社负责调换)

Preface

When work began on ROS at Stanford and Willow Garage in late 2007, there were big ambitious and big dreams, but no-one could have predicted that the project would grow so much and come to have such an impact. We set out to develop a common, open software platform on which everyone, everywhere can build their own robotics applications, and today, while there is always more to be done, we can say that we have moved a long way toward that goal.

We aimed initially for academic users, thinking primarily of graduate students in university labs who were doing cutting-edge research. The ROS user community grew quickly in academia, and soon there were new contributions to ROS coming in regularly from schools around the world. Robotics labs large and small started using ROS in their research, releasing demonstration videos mentioning ROS, and, most importantly, contributing enhancements and fixes that made the platform progressively better for everyone.

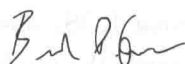
In 2010, Willow Garage released the first batch of PR2 mobile manipulation robots, sending them to top-tier research institutions on three continents. The result, as an entire cohort of graduate students had the opportunity to work with an advanced robot equipped with sophisticated open source software, was even greater participation in the ROS community from researchers. We then began to see what would become a wave of scholarly articles mentioning the PR2 robot and/or ROS. At the time of writing, Google Scholar indexes 2,000 citations of the original paper that we published on ROS.

In 2012, given the growth of the community, we decided to organize a ROS developers' conference, called ROSCon. Without knowing for sure whether anyone would travel to spend a weekend discussing open source robot software, we made all the arrangements and held ROSCon 2012 in St. Paul, Minnesota. To our surprise, over 200 people attended, and we enjoyed an educational and fun weekend of presentations, pitches, and discussions about

ROS and related topics. We have organized ROSCon each year since, with the latest edition, 2015, drawing a sell-out crowd of almost 400 people in Hamburg, Germany.

A significant part of the community growth in recent years has been in industry. There are now companies large and small using and relying on ROS, from early-stage research and development to full product deployment. We see ROS being used by industry in a wide array of applications, from smart factory and warehouse robots, to service robots, to self-driving cars, and increasingly, drones. We could not be more excited to see such a commercial impact of the work that our community has collectively produced over the past eight years.

It is impossible to know what will come next, for ROS specifically or for the field in general. But, for our part, we will keep improving, extending, and supporting the open source platform that has come to define how robotics work is done around the world. I hope that this book will serve to accelerate that work by recruiting more users and developers from China and elsewhere.



CEO of OSRF

Open Source Robotics Foundation

2015. 11. 25

前 言

ROS(Robot Operating System, 机器人操作系统)是一个开源的、通用的机器人软件开发平台,其前身是斯坦福人工智能实验室为了支持 STAIR(斯坦福大学人工智能机器人项目)而建立的交换庭(Switchyard)项目。从 2008 年开始,ROS 由位于硅谷的美国 Willow Garage 公司负责维护和运营,从而在全世界得到推广。2012 年,ROS 的维护和运营团队从 Willow Garage 独立出来,并且成立了一个非盈利组织 OSRF(Open Source Robotics Foundation, 开源机器人基金会)。凭借其无与伦比的通用性、开源性、复用性和社区性,经过近几年的发展,ROS 已经成为世界上应用范围最广的通用机器人软件开发平台。

上海硅步科学仪器有限公司是硅步国际有限公司的下属企业,专注于人工智能和机器人领域的技术引进、开发和转化。上海硅步从 2010 年就开始和美国 Willow Garage 公司合作,最早把 ROS 技术引入中国。现在,作为 OSRF(开源机器人基金会)的合作伙伴,上海硅步专注于基于 ROS 的各种机器人和传感器的市场推广和技术支持,先后举办了多次 ROS 培训和学术研讨会,为 ROS 在中国的推广发挥了积极的作用。为了方便更多的人了解和学习 ROS,上海硅步科学仪器有限公司根据自己在 ROS 领域多年的技术积累,组织公司的研发队伍和国内同行专家,编写了本书。

本书由李振伟统稿并执笔第 1 章、第 2 章和第 3 章;第 4 章由于泽宽执笔;第 5 章由马庆华执笔;第 6 章由陈萌执笔。全书结构清晰、由浅入深,第 1 章至第 3 章为 ROS 入门知识,第 4 章至第 6 章为 ROS 在机器人应用中的实战案例。本书可用作从事机器人、人工智能、自动化等相关领域的高年级本科生、研究生和科研人员的参考书,也可供广大的 ROS 爱好者自学使用。

本书在编写和出版过程中,得到了《ROS By Example》的作者 Patrick Goebel 博士、《Robot Operating System(ROS)》的作者 Anis Koubaa 博士和国内的诸多专家学者的指导和帮助。此外,OSRF 的 CEO Brian Gerkey 博士在百忙之中为本书

撰写了序言。在此,向所有支持本书编写和出版的专家致以衷心的感谢。

限于作者的水平和经验,不足和疏漏之处在所难免,敬请广大读者和同行专家批评指正。

言 前

作者
2016年1月

第 1 章 ROS 概述	1
1.1 ROS 简介	1
1.1.1 ROS 的发展	1
1.1.2 ROS 基本概念	2
1.2 ROS 安装	3
1.2.1 安装 ROS	3
1.2.2 安装 Gazebo	4
1.2.3 使用 SNN/Git/Mercurial	5
1.3 ROS 命令介绍	6
1.3.1 功能包命令	6
1.3.2 ROS 核心命令	7
第 2 章 ROS 2.0 开发基础	10
2.1 ROS 2.0 简介	10
2.1.1 ROS 2.0 诞生的背景	10
2.1.2 ROS 2.0 中的 DDS	10
2.2 ROS 2.0 安装	15
2.3 ROS 2.0 应用实例	16
2.3.1 实时控制	16
2.3.2 ROS 1.0 功能包转换为 ROS 2.0	18
2.3.3 ROS 1.0 和 ROS 2.0 通信	28
第 3 章 ROS 仿真	30
3.1 RViz 仿真工具	30
3.1.1 安装 RViz	30

3.1.2	RViz 应用方法	31
3.1.3	RViz 仿真实例	36
3.2	Gazebo 仿真工具	42
3.2.1	Gazebo 使用方法	42
3.2.2	Gazebo 仿真模型	45
3.2.3	Gazebo 仿真实例	51
第 4 章	灵巧手抓取规划	53
4.1	灵巧手简介	53
4.2	创建灵巧手抓取规划功能包	54
4.2.1	Shadow 手模型描述	54
4.2.2	创建 Shadow 手功能包	55
4.2.3	创建灵巧手逆运动学插件	62
4.3	综合抓取仿真实验	72
第 5 章	移动机器人导航定位	79
5.1	准备工作	79
5.2	简单的路径规划仿真	80
5.3	TurtleBot 导航仿真	83
5.4	TurtleBot 导航定位实验	85
第 6 章	机械臂运动规划	88
6.1	MoveIt! 安装方法	88
6.2	MoveIt! 接口	89
6.3	Baxter 机器人功能包	98
6.4	创建 MoveIt! 配置功能包	98
6.5	机械臂运动规划仿真	111
6.5.1	机械臂关节空间规划	111
6.5.2	机械臂笛卡尔空间规划	114
6.5.3	机械臂避障仿真	123
6.5.4	双臂协调运动	123
6.5.5	Baxter 机器人双臂操作	132
参考文献	139

第 1 章 ROS 概述

本章首先简单介绍一下 ROS 的发展以及特点,使读者能够初步了解 ROS。然后介绍 ROS 安装及其基本命令。

1.1 ROS 简介

1.1.1 ROS 的发展

随着机器人技术的快速发展,人们对开源、代码复用以及模块化设计的需求越来越强烈,但是已有的开源机器人系统不能很好地满足人们的需求。2010 年 Willow Garage 公司发布了开源机器人操作系统 ROS(robot operating system),此后 ROS 系统得到了快速的发展。2015 年,ROS 的版本已经更新到了 Jade,之前的版本包括 Electric、Fuerte、Groovy、Hydro、Indigo,而且越来越多的机器人使用 ROS 系统进行开发,如图 1-1 所示。

ROS 操作系统起源于 2007 年斯坦福大学人工智能实验室的项目与机器人技术公司 Willow Garage 的个人机器人项目(Personal Robots Program)之间的合作,2008 年之后就由 Willow Garage 来进行推动。ROS 是一种开源的机器人后操作系统,或者说次级操作系统。它能提供类似操作系统所提供的功能,例如硬件抽象描述、底层驱动程序管理、共用功能的执行、程序间的消息传递、程序发行包管理,它也提供一些工具程序和库用于获取、建立、编写和运行多机整合的程序。

ROS 是一种分布式的处理框架,这使得可执行文件能够被独立设计,并且在运行时松散耦合。ROS 可以分成两层,下层是上述的操作系统层,上层则是广大用户群贡献的能够实现不同功能的功能包,例如机械臂运动规划、自主导航定位、传感器插件、仿真工具等。

ROS 的主要特点为:

① 精简与集成:ROS 建立的系统具有模块化的特点,各模块中的代码可以单独编译,而且编译使用的 CMake 工具可以很容易地实现精简的理念。ROS

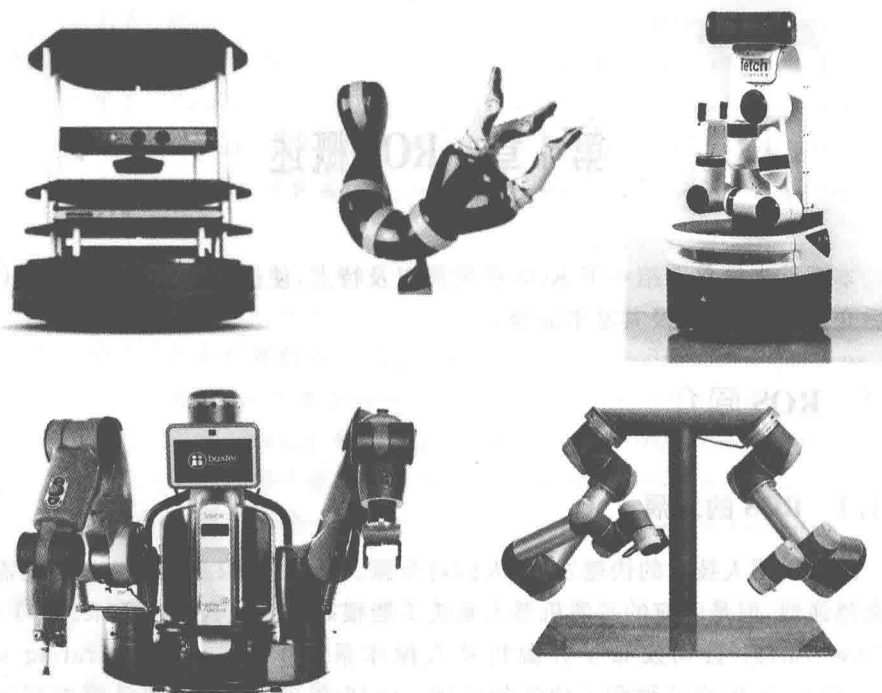


图 1-1 部分支持 ROS 的机器人

集成了很多现在已经存在的开源库,例如 OpenCV、OpenRAVE、Player 等。

② 支持多种编程语言:为了满足不同编程者的需求,ROS 采用了语言中立性的框架结构,即不依赖某一种编程语言。ROS 支持的语言包括 C++、Python、Octave 和 LISP 等。

③ 免费并且开源:ROS 遵循 BSD 协议,对个人、商业应用以及修改都是免费的。

④ 点对点设计:ROS 的点对点设计以及服务和节点管理等机制可以分散由计算机视觉和语音识别等功能带来的实时计算压力。

1.1.2 ROS 基本概念

为了更好地学习 ROS,需要先理解几个概念:

① 功能包(package):功能包是 ROS 中组织软件的主要形式,可以编写代码并进行编译、执行等操作。一个功能包一般包含程序文件、编译描述文件、配置文件等。

② 功能包集(stack):功能包集是实现某种功能的多个功能包的集合,可提供更高级的功能。每一个功能包集都带有相关版本号,是 ROS 软件发布的主要形式。

③ 节点(node):节点是一个可执行文件,多个节点可实现复杂的功能。程序文件只有转换为可执行文件,才可以在 ROS 中运行。

④ 主题(Topic)和服务(Service):节点之间的通信方式主要包括主题和服务两种。主题只能实现节点之间的单向通信,而服务是双向通信,包括请求(request)和响应(response)。

⑤ 消息(msg):消息是指通信的具体内容,每一个消息都有一个固定的数据结构,支持标准的原始数据类型,包括整形、浮点型、布尔型等。

1.2 ROS 安装

1.2.1 安装 ROS

虽然目前 ROS 已经有多个版本,但它们的安装过程类似。下面将以 Indigo 版本为例,介绍 ROS 的安装方法。Ubuntu 的版本为 64 位的 14.04 LTS。

(1) 配置 Ubuntu 系统

打开 ubuntu 系统的软件中心,并在菜单栏中选择 Edit—Software Sources。在 Ubuntu Software 一栏选中“restricted”、“universe”、“multiverse”。

(2) 配置 sources.list

配置 sources.list 使得计算机能够从 packages.ros.org 接收软件。

对于 Ubuntu 14.04:

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

(3) 安装 keys

```
$ sudo apt-key adv-keyserver hkp://pool.sks-keyservers.net --recv-key 0xB01FA116
```

(4) 安装

重新定向 ROS 服务器:

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

根据不同的功能,ROS 有如下四种安装:

① 桌面完整版安装(推荐):ROS、rqt、rviz、robot_generic 库、2D/3D 仿真器、导航以及 2D/3D 感知。

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

② 桌面安装:ROS、rqt、rviz、robot_generic 库。

```
$ sudo apt-get install ros-indigo-desktop
```

③ ROS-Base:ROS 的主要功能包、build 和 communication 库。

```
$ sudo apt-get install ros-indigo-ros-base
```

④ 独立的功能包:安装特定的 ROS 功能包。

```
$ sudo apt-get install ros-indigo-PACKAGE
```

安装完成以后,可以使用以下命令来查看已安装的功能包:

```
$ apt-cache search ros-groovy
```

(5) 初始化 rosdep

使用 ROS 之前,需要初始化 rosdep。

```
$ sudo rosdep init
```

```
$ rosdep update
```

(6) 环境设置

环境设置对操作 ROS 功能包有重要作用。运行下面的命令,为 Indigo 版本设置环境:

```
$ source /opt/ros/indigo/setup.bash
```

需要说明的是:每打开一个新的终端窗口都需运行上面的命令。为了实现环境的自动设置,可以将上面的命令写入 .bashrc 文件。运行下面的命令:

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

```
$ source ~/.bashrc
```

(7) 安装 rosinstall

rosinstall 是 ROS 中一个经常使用的命令行工具,它可以很容易地为 ROS 功能包下载 source 树。运行下面的命令:

```
$ sudo apt-get install python-roinstall
```

1.2.2 安装 Gazebo

Gazebo 是一种基于 ODE 物理引擎的仿真软件,可实现复杂的动力学/运动学仿真。下面将介绍如何安装 Gazebo。

对于 Groovy 以后的版本(Hydro、Indigo),ROS 所使用的 Gazebo 是一个不依赖于 ROS、独立的仿真软件。在 Indigo 中安装 Gazebo 的过程如下:

(1) 安装 keys

```
$ wget http://packages.osrfoundation.org/gazebo.key -O- | sudo apt-key add -
```

(2) 安装 Gazebo

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

```
$ sudo apt-get install gazebo
```

(3) 运行 Gazebo

安装完成之后,运行命令 `gzserver`,启动 Gazebo 服务器。然后打开另一个终端窗口,运行命令 `gzclient`,启动 Gazebo 客户端,运行结果将出现 Gazebo 的 GUI。也可直接运行命令 `gazebo`,以启动 Gazebo 服务器和客户端。

Gazebo 在运行时需要占用一定的系统资源。为了减小 CPU 的负载,Gazebo 利用了 CPU 的硬件加速。然而对于一些电脑,硬件加速可能导致 Gazebo 崩溃。在硬件允许的情况下,最好打开硬件加速:

```
$ unset LIBGL_ALWAYS_SOFTWARE
```

如果出现 Gazebo 崩溃的情况,可以尝试重新启动 Gazebo。如果还是失败,则关闭所有正在运行的 ROS 进程,包括 `roscore`,然后再重新启动。

在以上尝试都失败的情况下,也可以关闭硬件加速:

```
$ export LIBGL_ALWAYS_SOFTWARE=1
```

虽然关闭硬件加速可以解决 Gazebo 崩溃的问题,但将占用较多的 CPU 资源。

为了建立 ROS 和 Gazebo 之间的联系,需要安装与 Gazebo 相关的 ROS 功能包。下面是在 Indigo 中安装与 Gazebo 相关的功能包:

```
$ sudo apt-get install ros-indigo-gazebo-ros \
ros-indigo-gazebo-ros-pkgs ros-indigo-gazebo-msgs \
ros-indigo-gazebo-plugins ros-indigo-gazebo-ros-control
```

运行 `gazebo_ros` 功能包中的 `empty_world.launch` 文件:

```
$ roslaunch gazebo_ros empty_world.launch
```

若出现 Gazebo 的 GUI,则说明功能包安装成功。

1.2.3 使用 SVN/Git/Mercurial 安装功能包

目前三个主要的代码库分别为 SVN、Git 和 Mercurial。为了使用这三个代码库,需要运行下面的命令进行安装:

```
$ sudo apt-get install git subversion mercurial
```

(1) SVN

假设功能包所在的网址为: `http://repository/svn/package_name`,运行下面的命令下载功能包:

```
$ svn checkout http://repository/svn/package_name
```

```
$ cd package_name
```

```
$ rosmake
```

(2) Git

假设功能包所在的网址为: `git://repository/package_name`, 运行下面的命令下载功能包:

```
$ git clone git://repository/package_name
```

```
$ cd package_name
```

```
$ rosmake
```

(3) Mercurial

假设功能包所在的网址为: `http://repository/package_name`, 运行下面的命令下载功能包:

```
$ hg clone http://repository/package_name
```

```
$ cd package_name
```

```
$ rosmake
```

1.3 ROS 命令介绍

1.3.1 功能包命令

(1) rospack(ros+pack(age))

rospack 用于获取功能包的相关信息, 其命令格式为: `rospack<command> [package_name]`。

例如:

① `rospack help`, 输出 rospack 命令的使用方法。

② `rospack find [package_name]`, 查找功能包, 返回的是功能包的绝对路径。

③ `rospack depends [package_name]`, 输出功能包的所有依赖项。

(2) rosstack(ros+stack)

rosstack 获取功能包集的相关信息。例如:

① `rosstack help`, 输出 rosstack 的帮助信息, 即 rosstack 命令的使用方法。

② `rosstack find [stack_name]`, 查找功能包集, 返回功能包集的绝对路径。

③ `rosstack depends [stack_name]`, 输出功能包集的所有依赖项。

(3) rosls(ros+ls)

rosls 是 rosbash 套件的一部分, 它可通过功能包的名称列出其包含的文

件/文件夹,而不必使用绝对路径,命令格式为:roslns [package_name/[sub-dir]]。

(4) roscd(ros+cd)

roscd 改变当前目录到指定的功能包/功能包集,命令格式为:roscd [package_name /[subdir]]

(5) catkin_create_pkg

创建 ROS 功能包,其命令格式为:

```
$ catkin_create_pkg [package_name] [depend1] [depend2] [depend3]...
```

其中[package_name]后面的部分是新建功能包所依赖的功能包,其作用相当于一个程序文件所包含的头文件。一个功能包可以有多个依赖功能包。

(6) catkin_make

catkin_make 的作用是编译 ROS 功能包。

1.3.2 ROS 核心命令

(1) roscore(ros+core)

对于 ROS 1.0,在运行 ROS 节点之前,必须先运行 roscore 命令。

(2) rosrn (ros+run)

rosrun 运行 ROS 节点,命令格式为:rosrun [package_name] [node_name]。

(3) rosnode(ros+node)

rosnode 可以显示正在运行的 ROS 节点信息。rosnode 的常用命令有如下几种:

① rosnode list,列出当前正在运行的节点。

② rosnode info [node_name],输出节点的信息。

③ rosnode kill [node_name],结束一个正在运行的节点,例如 rosnode kill-a,结束所有的节点。

(4) rosmsg/rossrv(ros+msg/ros+srv)

rosmsg 显示消息数据结构的定义,rossrv 显示服务数据结构的定义。可以运行 rosmsg-h (rossrv -h)查看 rosmsg(rossrv)命令的使用方法。以 rosmsg 为例,运行 rosmsg-h,可以看到 rosmsg 有如下几种命令:

rosmsg show 显示消息中各个变量的定义

rosmsg list 列出 ROS 中所有的消息

rosmsg md5 显示消息的 md5 值

rosmsg package 列出功能包中的所有消息

`rosmmsg packages` 列出具有某个消息的所有功能包

(5) `rostopic(ros+topic)`

`rostopic` 查看节点的主题信息,运行 `rostopic-h` 可以获得该命令的使用方法,如下:

Commands:

`rostopic bwdisplay` bandwidth used by topic

`rostopic echoprint` messages to screen

`rostopic findfind` topics by type

`rostopic hzdisplay` publishing rate of topic

`rostopic infoprint` information about active topic

`rostopic listlist` active topics

`rostopic pubpublish` data to topic

`rostopic typeprint` topic type

◇ `rostopic bw [topic]`

显示主题的带宽;

◇ `rostopic echo [topic]`

输出主题发布的数据;

◇ `rostopic find`

通过类型查找主题;

◇ `rostopic hz [topic]`

输出主题发布的频率;

◇ `rostopic info [topic]`

输出主题的基本信息,包括消息类型、发布节点和订阅节点;

◇ `rostopic list`

输出当前活动的主题;

◇ `rostopic pub`

发布数据到主题;

◇ `rostopic type [topic]`

输出主题的消息类型。

(6) `rosservice`

`rosservice` 是一个应用于服务的命令。运行命令 `rosservice-h`,可以获得 `rosservice` 的帮助信息。`rosservice` 有如下几种使用方法:

`rosservice type` 输出服务的类型

`rosservice find` 通过服务类型查找服务

`rosservice uri` 输出服务 ROSRPC uri

`rosservice list` 输出当前活动服务的的信息

`rosservice call` 请求服务

(7) `rosparam`

`rosparam` 可用来保存和设置 ROS 参数服务器(Parameter Server)中的数据。参数服务器可以存储整型、浮点型、布尔型、字典及列表。`rosparam` 的语法格式采用了 YAML 语言。YAML 语言中:1 表示整型,1.0 表示浮点型,one 表