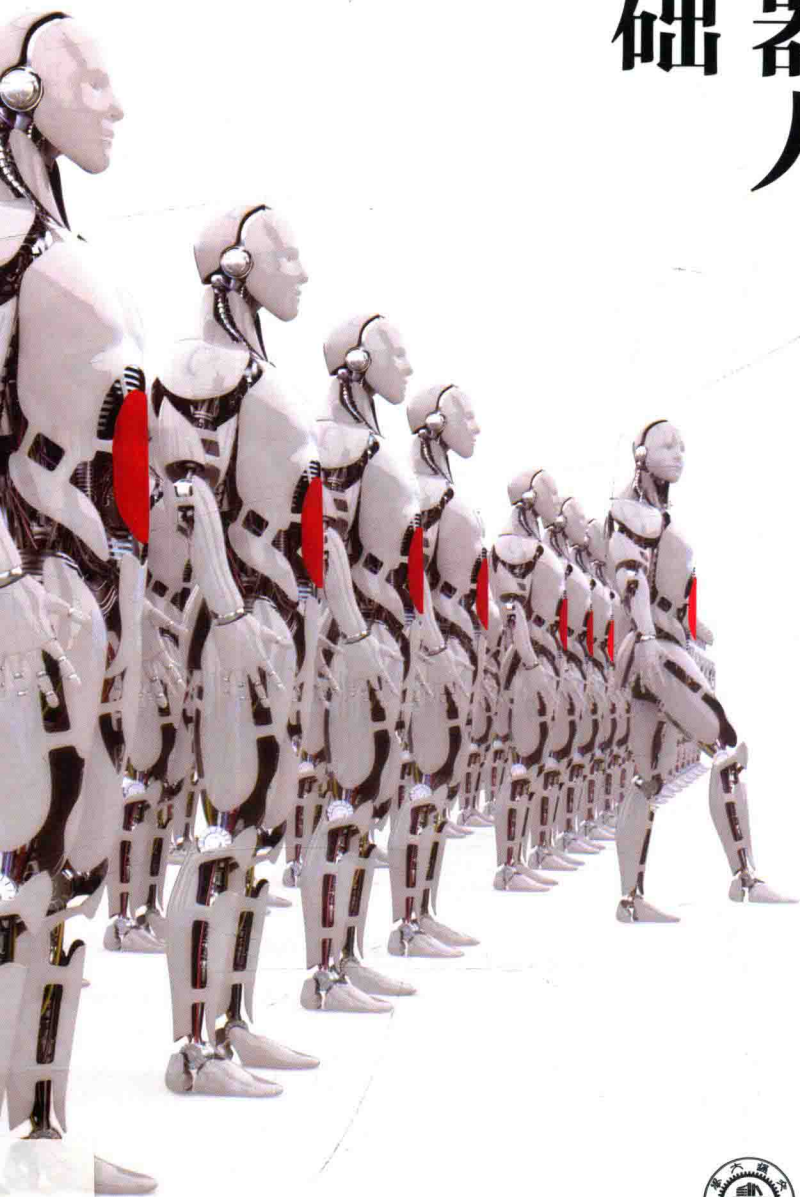
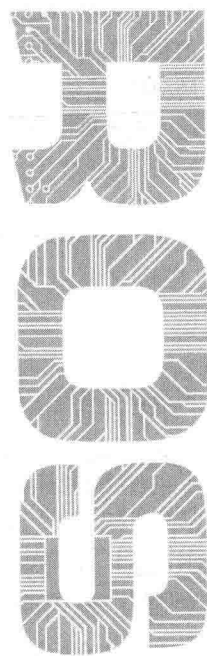


# 开源机器人 控制基础

陈金宝 韩冬  
聂宏 陈萌  
◎ 编著



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS



# 开源机器人 控制基础

陈金宝 韩冬  
聂宏 陈萌

◎ 编著



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS

## 内容提要

本书全面介绍了该操作系统的软件架构、编程方法以及智能机器人控制技术。主要内容包括：ROS 以及相关软件的安装方法、基本命令的使用、ROS 开发基础、ROS 中的仿真工具 Rviz 和 Gazebo、机器人抓取操作仿真、移动机器人定位导航仿真与实验、Moveit! 运动规划、ROS 通用硬件接口以及具有实时性的 ROS 2.0。

本书可作为智能机器人研究及自动化相关方向的科研人员与工程人员的参考书。

## 图书在版编目(CIP)数据

ROS 开源机器人控制基础 / 陈金宝等编著. —上海:  
上海交通大学出版社, 2016  
ISBN 978-7-313-14075-3

I. ①R… II. ①陈… III. ①机器人控制 IV.  
①TP24

中国版本图书馆 CIP 数据核字(2015)第 278681 号

## ROS 开源机器人控制基础

编 著: 陈金宝 韩冬 聂宏 陈萌

出版发行: 上海交通大学出版社

邮政编码: 200030

出 版 人: 韩建民

印 制: 昆山市亭林印刷有限责任公司

开 本: 787 mm×1092 mm 1/16

字 数: 405 千字

版 次: 2016 年 1 月第 1 版

书 号: ISBN 978-7-313-14075-3/TP

定 价: 52.00 元

地 址: 上海市番禺路 951 号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 17.75

印 次: 2016 年 1 月第 1 次印刷

版权所有 侵权必究

告读者: 如发现本书有印装质量问题请与印刷厂质量科联系

联系电话: 0512-57751097

# 前 言

目前机器人技术处于快速发展的时期,其应用领域涉及公共服务、工业、航空航天、军事等,产生了巨大的经济效益和社会效益。然而机器人技术存在重复开发的问题,降低了机器人的开发效率,限制了机器人技术的进一步发展。为了提高机器人代码复用和模块化设计水平,需采用具有更好通用性的开源机器人系统。2010年 Willow Garage 公司发布了开源机器人操作系统 ROS(Robot Operating System)。ROS 具有点对点设计、支持多种编程语言、免费开源等优点,因此得到了快速的推广和发展,目前已有几十种先进机器人支持 ROS,例如 PR2、Shadow 手、Baxter 等。

本书根据作者在 ROS 开发中遇到的问题以及经验,深入浅出地介绍了 ROS 的基本使用方法以及实际应用,涉及移动机器人自主导航、灵巧机械手抓取操作、多自由度机械臂运动规划等,可帮助不同领域的机器人开发人员了解并熟练使用 ROS。

本书的创新之处是以 ROS 开源系统为平台,系统地介绍不同类型机器人的智能控制方法,并提供了具体实例以及详细分析,主要内容分为八章。其中第一章主要介绍 Ubuntu 系统和 ROS 的安装及其基本命令,使读者初步了解 ROS 的特点。第二章主要介绍 ROS 的基本使用方法。第三章介绍 ROS 中两种常用的仿真工具: Rviz 和 Gazebo,涉及机器人仿真模型的创建以及添加 ROS 控制器。第四章主要介绍利用 ROS 实现灵巧机械手的智能抓取。第五章主要介绍利用 ROS 实现移动机器人的自主导航与避障。第六章介绍采用 MoveIt! 强大的功能实现多自由度机械臂的运动规划。第七章主要介绍通用型的 ROS 硬件接口模块。第八章介绍了最新推出的 ROS 2.0,ROS 2.0 可实现实时控制,在航空航天、自动化生产等实时性要求高的领域具有广阔的应用前景。

书中,第一、二章由韩冬、陈金宝、聂宏执笔;第三、五章由韩冬、陈萌执笔;第四、六章由韩冬、陈金宝、陈萌执笔;第七章由韩冬、陈金宝执笔;第八章由韩冬执笔。

本书结构清晰、合理,由易及难,通过大量的实例循序渐进地引导读者熟悉 ROS 的使用方法,对广大从事机器人相关领域开发的研究人员,不失为一本重要的参考书。

在本书编写过程中得到了上海硅步科学仪器有限公司常峰总经理以及马庆华工程师的支持和帮助,并提供了实验所需的机器人,编者在此深表谢意。

由于编者水平有限,书中存在的错误和欠缺之处,敬请读者批评指正。

编 者

# 目录 ■■■■

■ 1	ROS 简介 .....	001
1.1	ROS 概述 .....	001
1.2	Ubuntu 系统简介 .....	003
1.3	ROS 安装 .....	006
1.4	ROS 的基本命令 .....	013
■ 2	ROS 开发基础 .....	023
2.1	ROS 功能包 .....	023
2.2	节点(nodes) .....	027
2.3	消息(messages) .....	027
2.4	服务文件(srv) .....	029
2.5	主题(topics) .....	032
2.6	服务(services) .....	039
2.7	launch 文件 .....	044
2.8	rqt .....	047
2.9	ROS 中的运动学/动力学库 .....	053
2.10	ROS 与 Matlab .....	057
■ 3	ROS 中的仿真工具 .....	064
3.1	rviz .....	064
3.2	Gazebo .....	078
■ 4	机器人抓取操作仿真 .....	102
4.1	R2 机器人简介 .....	102
4.2	功能包安装与编译 .....	102
4.3	仿真设置 .....	107

4.4	R2 机器人的控制 .....	110
4.5	添加自己的控制程序 .....	122
4.6	抓取操作仿真 .....	134
<b>■ 5</b>	<b>导航定位仿真与实验</b> .....	<b>150</b>
5.1	SLAM 和 ROS 中的导航定位功能包 .....	150
5.2	安装 rbx1 .....	151
5.3	路径规划 .....	152
5.4	导航定位仿真 .....	162
5.5	实际机器人的导航定位实验 .....	168
<b>■ 6</b>	<b>MoveIt! 运动规划</b> .....	<b>181</b>
6.1	MoveIt! 安装 .....	181
6.2	MoveIt! 接口 .....	182
6.3	安装 Baxter 机器人的功能包 .....	195
6.4	创建自己的 MoveIt! 功能包 .....	196
6.5	机械臂运动规划实例 .....	219
6.6	MoveIt! 总结 .....	251
<b>■ 7</b>	<b>ROS 硬件接口</b> .....	<b>252</b>
7.1	RoNex 简介 .....	252
7.2	RoNex 使用方法 .....	254
7.3	RoNex 总结 .....	266
<b>■ 8</b>	<b>实时系统——ROS 2.0</b> .....	<b>267</b>
8.1	ROS 2.0 概述 .....	267
8.2	安装 ROS 2.0 .....	269
8.3	ROS 2.0 实例 .....	270
<b>■</b>	<b>主要参考文献</b> .....	<b>278</b>

# 1 ROS 简介

本章简单介绍 ROS 的发展与特点,使读者能够初步了解 ROS。此外,介绍 Ubuntu 系统的使用方法和常用命令,以及 ROS 的安装和基本命令。

## 1.1 ROS 概述

### 1.1.1 ROS 简介

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),之后 ROS 系统得到了快速的推广和发展。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,而且已经有几十种先进的机器人使用 ROS 系统进行开发,如图 1-1 所示。

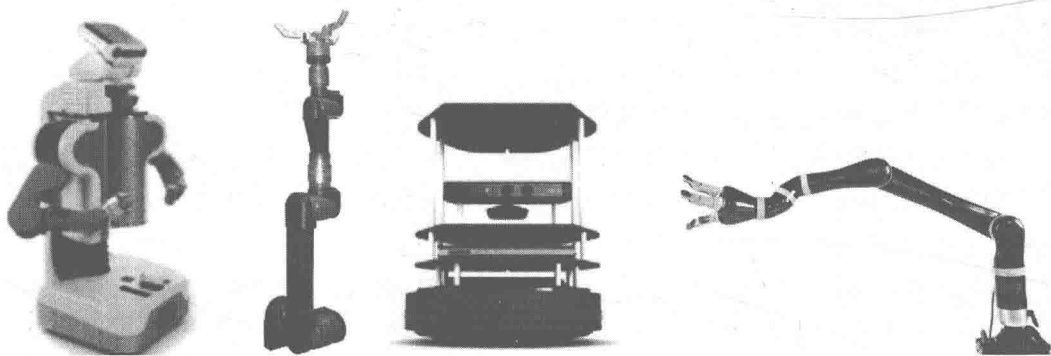


图 1-1 使用 ROS 开发的机器人

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,2010 年由 Willow Garage 公司正式发布。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。这些过程可以封装到功能包(packages)或功能包集(stacks)中,以便于共享和复用,这也是 ROS 的主要设计目标。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,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) 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,能够适应多机器人系统遇到的挑战。

(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。目前 ROS 支持的语言包括 C++、Python、Octave 和 LISP 等。

(3) 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 集成了很多现在已经存在的开源库,例如在机器视觉算法方面,ROS 已经集成了 OpenCV 库;在规划算法方面,ROS 已经与 OpenRAVE 实现集成;在驱动、运动控制、仿真方面与 Player 实现集成。

(4) 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,这也是促进 ROS 快速发展的主要因素之一。

### 1.1.2 ROS 中的几个概念

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,其作用相当于一个 C++ 工程,在该“工程”中我们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,它可以提供更高级的功能,例如“导航功能包集”。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): 节点实际上是一个可执行文件,它位于功能包中,是实现功能的最小单位。程序文件(例如, .cpp 文件和 .py 文件等)只有转换为可执行文件,才可以在 ROS 中运行。一个功能包中可以有多个节点。

(4) 主题(Topic)和服务(Service): 单个节点能够实现的功能比较简单,而多个节点联合起来就可以实现复杂的功能。这涉及节点之间的通信,其通信方式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)、响应(Response)。

(5) 消息(Msg): 消息指的是节点之间通信的内容,每一个消息都有一个固定的数据结构,它支持标准的原始数据类型,包括:整型、浮点型、布尔型等。



### 1.1.3 ROS 的网站

可以通过登录 ROS 的官网获取 ROS 最新的学习资料和资源。ROS 的官网为：<http://wiki.ros.org/>（若该网站无法登录，我们也可以登录 <http://mirror-ap.wiki.ros.org/>）。ROS 官网上的内容包括 ROS 的安装、学习教程、使用 ROS 开发的机器人、书籍、论文等。

## 1.2 Ubuntu 系统简介

ROS 支持最好的操作系统是 Ubuntu 系统。Ubuntu 系统是一个以桌面应用为主的 Linux 操作系统，它基于 Debian 发行版和 GNOME 桌面环境。相对于其他的 Linux 系统，Ubuntu 系统具有友好的操作界面，且简单易学。目前 Ubuntu 有多个版本，本书主要使用的是 Ubuntu 12.04。关于 Ubuntu 系统的安装，网上有很多教程，读者可参考进行安装，这里不再赘述。下面将简单介绍一下 Ubuntu 系统的使用以及常用命令。

### 1.2.1 Ubuntu 系统的使用

Ubuntu 系统的桌面如图 1-2 所示，其中主文件夹类似于 Windows 系统中“我的电脑”；在软件中心可以查看 Ubuntu 系统安装的软件信息，也可安装/卸载软件；在 Dash 主页中可以搜索文件、文件夹以及应用软件等，例如输入终端或 terminal 可以搜索到 Ubuntu 系统的终端。终端是 Ubuntu 系统的命令窗口，我们也可以采用快捷键 Ctrl+Alt+T 来打开终端窗口。



图 1-2 Ubuntu 系统的桌面

## 1.2.2 Ubuntu 系统中常用的命令

### 1) 目录操作

命令名称	解 释	举 例
ls	(list) 列出当前目录下的文件(不包括隐含文件)	
ls -a	列出当前目录下的所有文件,包括隐含文件	
mkdir <i>directory</i>	(make <i>directory</i> ) 新建一个目录 <i>directory</i>	mkdir ~/ros_workspace, ~表示用户主目录
rmdir <i>directory</i>	(remove <i>directory</i> ) 删除空目录 <i>directory</i>	
cd <i>directory</i>	(change <i>directory</i> ) 进入 <i>directory</i> 目录	cd ~/ros_workspace
cd	进入主目录	
cd ~	进入主目录	
cd ..	进入当前目录的上一级目录	
pwd	(print working directory) 显示当前目录的路径	

### 2) 文件操作

命令名称	解 释	举 例
cp <i>file1 file2</i>	(copy) 在当前目录下,复制 <i>file1</i> 并命名为 <i>file2</i>	cd ~/ros_workspace cp test1.txt new_test.txt 在 ~/ros_workspace 目录下复制 test1.txt 文件,并将新文件命名为 new_test.txt
mv <i>file1 file2</i>	(move)将 <i>file1</i> 文件重命名为 <i>file2</i>	
mv <i>file1 directory</i>	将 <i>file1</i> 从当前目录移动到 <i>directory</i> 目录下	cd ~/ros_workspace mv test1.txt ~ 将~/ros_workspace 目录中 test1.txt 文件移动到主目录
rm <i>file</i>	(remove)删除文件	cd ~/ros_workspace rm new_test.txt
grep 'keyword' <i>file</i>	在 <i>file</i> 文件中查找关键词 'keyword'	cd ~/ros_workspace grep i test1.txt 查找 test1.txt 文件中 'i' 变量

### 3) 修改文件权限

Ubuntu 系统中文件的调用权限有三种级别：所有者 (user)、群组 (group)、其他 (other)。一般情况下，我们可以在文件/文件夹的属性中直接修改操作权限，如图 1-3 所示。但是如果文件/文件夹的属性显示当前用户没有读写权力时，则无法修改操作权限，如图 1-4 所示。



图 1-3 可修改权限



图 1-4 无法修改权限

当无法直接修改权限时，我们可以使用 `chmod` (changing a file mode) 命令修改文件的权限。`chmod` 的常用用法：

- `sudo chmod 600 × × ×` (只有所有者有读和写的权限)
- `sudo chmod 644 × × ×` (所有者有读和写的权限, 组用户只有读的权限)
- `sudo chmod 666 × × ×` (每个人都有读和写的权限)
- `sudo chmod 700 × × ×` (只有所有者有读和写以及执行的权限)
- `sudo chmod 777 × × ×` (每个人都有读和写以及执行的权限)

其中 `×××` 指文件名，如果是文件夹，则需要加 `ld`。上面的三位数的每一位都表示一个用户类型 (所有者、群组、其他) 的权限设置，每一位的取值是 0~7，即二进制的 `[000]~[111]`，其中 0 `[000]` 表示无任何权限，4 `[100]` 表示只读权限，6 `[110]` 表示可读可写权限，7 `[111]` 表示可读可写可执行。例如 `sudo chmod 777 * -R` (`-R` 表示递归处理，`*` 代表所有文件)，表示任何人都可以读写、执行当前目录下的所有文件/文件夹。

### 4) 操作软件包

`sudo apt-get install softname` 安装软件包

```

apt-cache search softname  搜寻软件包
sudo apt-get install softname -- reinstall  重新安装软件包
sudo apt-get remove softname  卸载软件包
sudo apt-get remove -- purge softname  卸载软件包并清除配置
apt-get update  更新源
apt-get upgrade  更新已经安装的软件包
make 编译

```

## 1.3 ROS 安装

### 1.3.1 安装 ROS(Groovy)

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 Groovy 版本为例,介绍 ROS 的安装方法。

#### 1) 配置 Ubuntu 系统

打开 Ubuntu 系统的软件中心,如图 1-5 所示。



图 1-5 Ubuntu 的软件中心

将鼠标移到软件中心界面的最上端,出现菜单栏,并选择 Edit-Software Sources,出现如图 1-6 所示的对话框,在 Ubuntu Software 一栏选中“restricted”“universe”“multiverse”。

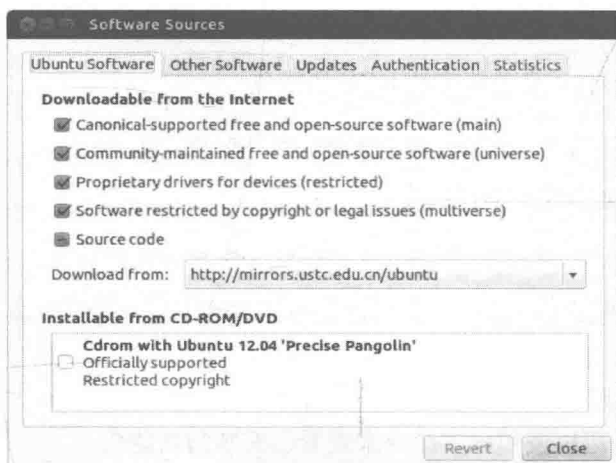


图 1-6 Ubuntu 的源设置

## 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 12.04 (Precise):

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

对于 Ubuntu 12.10 (Quantal):

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

## 3) 安装 keys

```
$ wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.
key-0 - | sudo apt-key add -
```

## 4) 安装 ROS

重新定向 ROS 服务器:

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

根据不同的功能,ROS 有如下四种安装方式:

(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-groovy-desktop-full
```

(2) 桌面安装: ROS、rqt、rviz、robot\_generic 库。

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

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

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

(4) 独立的功能包：安装特定的 ROS 功能包。

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

在安装中可能会出现如下的错误：

```
E: Unable to locate package ros-goovy-desktop-full
```

解决的方法：更新 Ubuntu 的源。首先在终端窗口运行命令：`sudo gedit /etc/apt/sources.list`，打开源列表文件；其次将新的源复制到该文件，并保存；然后执行 `sudo apt-get update` 命令，更新源；最后重新运行安装命令。

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。一般出现“无法找到或使用功能包”的问题都是环境设置不正确造成的。设置环境的方法是：`source` 环境设置文件(`setup.*sh`)，更新环境变量(如 `ROS_ROOT`、`ROS_PACKAGE_PATH`)。`setup.*sh` 文件存在于三种不同的地方：

(1) 由功能包管理器安装的 ROS 功能包提供 `setup.*sh` 文件。

(2) 使用 `rosws` 命令创建的 `rosws` 工作空间提供 `setup.*sh` 文件。

(3) 编译 `catkin` 功能包可生成 `setup.*sh` 文件。

前面我们已经在 Ubuntu 系统中安装了 ROS，可以在 `/opt/ros/<distro>` 目录下找到 `setup.*sh` 文件，其中 `<distro>` 表示 ROS 的版本。运行下面的命令，为 Groovy 版本设置环境：

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

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

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

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

如果安装了多个版本的 ROS，当由一个版本转换到另一个版本时，可以修改 `.bashrc` 文件。例如由 Groovy 版本转换到 Hydro 版本：首先运行 `gedit ~/.bashrc`，在打开的 `.bashrc` 文件中，将 `source /opt/ros/groovy/setup.bash` 改为 `source /opt/ros/hydro/setup.bash`；然后保存 `.bashrc` 文件并关掉；最后运行 `source ~/.bashrc` 即可转换到

Hydro 版本。查看系统当前使用的 ROS 版本,可以运行 `export | grep ROS` 命令。

### 7) 安装 `roscpp`

`roscpp` 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。执行下面的命令:

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

ROS 的安装目录是文件系统中的 `/opt/ros`,读者可以查看其中的功能包。如果要卸载 ROS,可以运行命令: `$ sudo apt-get purge ros-<distro>-*`,其中 `<distro>` 表示 ROS 的版本名。

## 1.3.2 安装 Gazebo

Gazebo 是 ROS 中的物理仿真环境,它基于 ODE 物理引擎,在机器人仿真中具有重要作用。下面将介绍如何安装 Gazebo。

在 Groovy 及之前版本中,Gazebo 均以 `simulator_gazebo` 功能包的形式安装于 ROS 中。因此首先需要安装 `simulator_gazebo` 功能包:

```
$ sudo apt-get install ros-%YOUR_ROS_DISTRO%-simulator-gazebo (%YOUR_ROS_DISTRO% 是 ROS 的版本: Electric、Fuerte、Groovy)
```

然后设置 ROS 的环境变量:

```
$ source /opt/ros/%YOUR_ROS_DISTRO%/setup.bash
```

最后我们可以执行下面的命令来启动 Gazebo:

```
$ roslaunch gazebo_worlds empty_world.launch
```

如果 Gazebo 安装成功,则会出现图 1-7 的界面。



图 1-7 Gazebo

对于 Groovy 以后的版本 (Hydro、Indigo 等), ROS 所使用的 Gazebo 是一个不依赖于 ROS 的独立仿真软件。在 Hydro 中可以安装 1.9. 版本的 Gazebo。安装过程如下:

### 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, 执行 `gzserver` 命令, 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, 如图 1-7 所示。我们也可直接执行命令 `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。

上面已经提到, 对于 Groovy 以后的版本, ROS 所使用的 Gazebo 是一个独立的仿真软件。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 Hydro 中安装与 Gazebo 相关的功能包:

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

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

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

如果出现 Gazebo 的 GUI, 则说明功能包安装成功。

## 1.3.3 安装 PR2 机器人功能包

PR2 机器人是较早使用 ROS 进行开发的智能机器人。目前, PR2 的 ROS 功能包涉及机械臂的运动规划、控制、逆运动学求解、碰撞检测以及自主导航等, 既可用于仿真也可



用于实物控制。因此,PR2 的功能包对于 ROS 的学习者和开发人员具有较高的参考价值和应用价值。下面我们将介绍如何安装 PR2 机器人的功能包。

#### 1) 配置 sources.list

对于 Ubuntu 12.04 (Precise):

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

对于 Ubuntu 12.10 (Quantal):

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

#### 2) 安装 keys

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

#### 3) 安装 PRI

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

```
$ sudo apt-get install ros-groovy-pr2-desktop
```

安装完成后,可以执行下面的命令,以验证 PR2 的功能包是否安装成功。

```
$ roslaunch pr2_gazebo pr2_empty_world.launch
```

运行结果如图 1-8 所示。

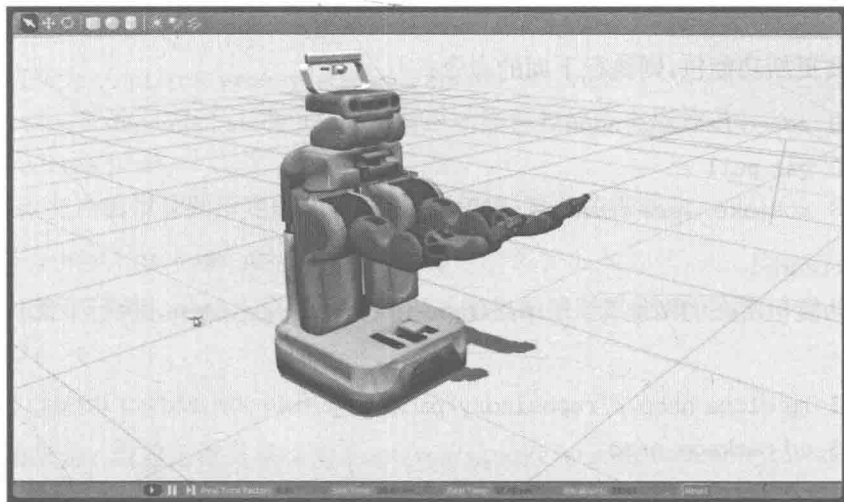


图 1-8 PR2 机器人

### 1.3.4 使用 SVN/Git/Mercurial 安装功能包

上面使用 apt-get 命令安装 ROS 功能包,但该命令并不是万能的,有时需要从其他代