GaiTech
www.gaitech.net

上海硅步科学仪器
有限公司支持出版

ROS By Example (Volume 2)

Packages and Programs for Advanced Robot Behaviors

ROS进阶实例

[美] R. 帕特里克·戈贝尔 (R. Patrick Goebel) ◎ 著
[墨] J. 罗哈斯 (Juan Rojas)
刘振东　罗　盛　林晨宇　杨浦城　陈雅雪　黄镇杰　郭力维　李子然 ◎ 译

中山大學出版社
SUN YAT-SEN UNIVERSITY PRESS

GaiTech
www.gaitech.net

上海硅步科学仪器
有限公司支持出版

ROS By Example (Volume 2)

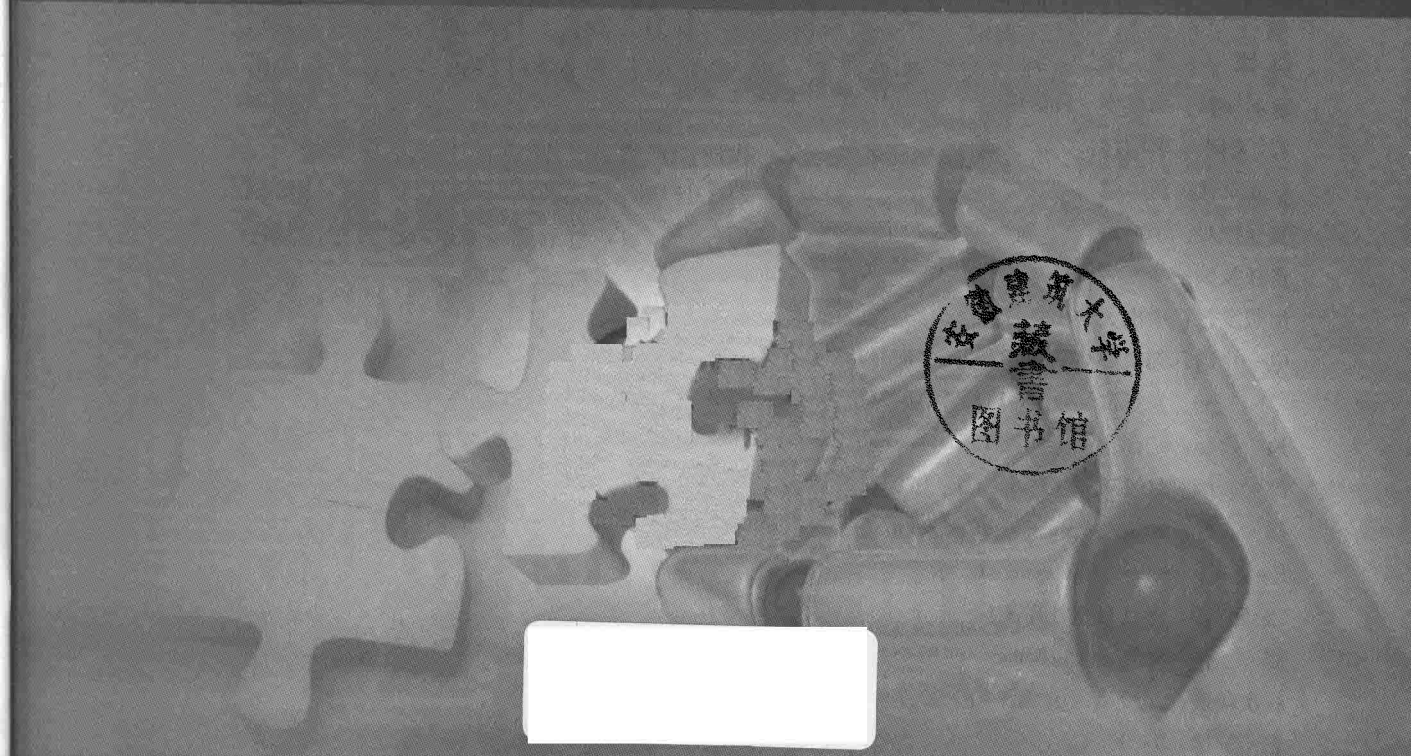
Packages and Programs for Advanced Robot Behaviors

ROS进阶实例

[美] R. 帕特里克·戈贝尔 (R. Patrick Goebel) ◎ 著

[墨] J. 罗哈斯 (Juan Rojas)

刘振东　罗　盛　林晨宇　杨浦城　陈雅雪　黄镇杰　郭力维　李子然 ◎ 译



中山大學出版社
SUN YAT-SEN UNIVERSITY PRESS
·广州·

图书在版编目（CIP）数据

ROS 进阶实例/（美）R. 帕特里克·戈贝尔（R. Patrick Goebel）著；（墨）J. 罗哈斯（Juan Rojas）等译. —广州：中山大学出版社，2017.4

ISBN 978-7-306-06030-3

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

中国版本图书馆 CIP 数据核字（2017）第 068188 号

出 版 人：徐　劲
策划编辑：周建华　黄浩佳
责任编辑：黄浩佳
封面设计：曾　斌
责任校对：李艳清
责任技编：何雅涛
出版发行：中山大学出版社
电　　话：编辑部 020-84110283，84111996，84111997，84113349
　　发行部 020-84111998，84111981，84111160
地　　址：广州市新港西路 135 号
邮　　编：510275　　传　　真：020-84036565
网　　址：http：// www. zsup. com. cn
E-mail：zdcbs@ mail. sysu. edu. cn
印 刷 者：湛江日报社印刷厂
规　　格：880mm×1230mm　1/16　　28.875 印张　　670 千字
版次印次：2017 年 4 月第 1 版　2017 年 4 月第 1 次印刷
定　　价：98.00 元

目　录

1 本卷范围

在第一卷（《ROS 入门实例》）中，我们学到了如何使用 ROS 的基本组件来对机器人进行编程，这些组件包括对可移动底座的控制、SLAM、机器人视觉（OpenCV，OpenNI 和一点 PCL）、语音识别和使用伺服电机来控制关节。在这一卷书中，我们将会面对更高级的 ROS 概念和一些对于真正的自动机器人行为编程有用的程序包，包括如下：

- 可执行任务管理器，例如：smach 和behavior trees。
- 使用URDF/Xacro 描述性语言来为你自己的机器人创建一个模型，这个模型包括一个可抬头摇头的头部、多关节机械臂和夹持器、可伸缩的躯干和安置像激光扫描仪那样的传感器。
- 配置arbotix 程序包来控制伺服电机。
- 使用 ROS 诊断程序包使你的机器人可以监测它自己的系统，例如电池水平和伺服电机温度。
- 使用dynamic reconfigure 来即时改变机器人的参数。
- 使用mux 和yocs 来复用 ROS 主题，使控制输入可以被优先考虑，并且这些控制输入不会互相矛盾。
- 使用AR 标签用于目标检测和追踪。
- 跟踪三维物体，并感知物体和机器人的空间关系。
- 使用新的MoveIt! 框架来控制多关节臂和夹持器，包括正向运动和反向运动的执行、碰撞避免、拾取、挑选和放置任务。
- 使用先进的Gazebo 模拟器来与模拟机器人和环境工作。
- 使用rosbridge，HTML 和 Javascript 为机器人创建一个基于 web 的图形用户界面。

注意：在 Gazebo 的那一章里，假定我们已经有一个像Kobuki 或者Fetch 那样的机器人模型，这些模型的开发者们提供了必要的 Gazebo 特性和插件来模拟机器人的物理特性。为一个新的机器人从头开始创建 worlds 或者 Gazebo 插件已经超出了这卷书的范围，但是这些在在线的Gazebo 教程中有介绍。

在这本书中，我们将要介绍的一些主题如果单独拿出来可以用一整本书去介绍，所以我们将重点放在关键的概念介绍上。如果读者需要探索进一步的细节，我们也提

供了额外的参考。如在第一卷，所有的代码示例都是用 Python 写的。这种语言往往比 C + +更容易入门。

一旦你掌握了本书中呈现过的概念，那你就能够为你的机器人创建一个 URDF 模型了。这个模型包括一个机械臂和一个可抬头或摇头的头部。然后编程使之进行一系列自动的任务和行为、监视它的子系统、确定控制输入的优先级以匹配当前状况、追踪三维物体、了解如何用碰撞感知逆运动学来为多关节臂和夹持器进行编程，并且使用网页浏览器编写 HTML/Javascript 接口来监视和控制你的机器人。

2 安装 ROS-BY-EXAMPLE 代码

请注意本书中的代码都是以 ROS Indigo 版本写的。虽然其中有一些代码能在更早的版本 ROS 中运行，但强烈建议读者使用 ROS Indigo 版本。特别是下面列出的依赖程序包和为机械臂导航的 MoveIt！样例代码程序是只适用于 Indigo 的。

在安装第二卷代码本身之前，如果我们先安装那些大部分以后会用到的 ROS 程序包的话，后续可以节省一些时间。（在本书中如需安装某单个程序包，安装指导同样会给出。）简单地复制粘贴以下命令（不包括 $ 符号）进终端来安装我们所需的 Debian 程序包。在每行命令末尾的 \ 字符让整个代码块对于 Linux 来说是作为一整行进行复制粘贴的：

```
$ sudo apt-get install ros-indigo-arbotix ros-indigo-openni-camera \
  ros-indigo-dynamixel-motor ros-indigo-rosbridge-suite \
  ros-indigo-mjpeg-server ros-indigo-rgbd-launch \
  ros-indigo-moveit-full ros-indigo-moveit-ikfast \
  ros-indigo-turtlebot-* ros-indigo-kobuki-* ros-indigo-moveit-python \
  python-pygraph python-pygraphviz python-easygui \
  mini-httpd ros-indigo-laser-pipeline ros-indigo-ar-track-alvar \
  ros-indigo-laser-filters ros-indigo-hokuyo-node \
  ros-indigo-depthimage-to-laserscan \
  ros-indigo-gazebo-ros ros-indigo-gazebo-ros-pkgs \
  ros-indigo-gazebo-msgs ros-indigo-gazebo-plugins \
  ros-indigo-gazebo-ros-control ros-indigo-cmake-modules \
  ros-indigo-kobuki-gazebo-plugins ros-indigo-kobuki-gazebo \
  ros-indigo-smach ros-indigo-smach-ros ros-indigo-grasping-msgs \
  ros-indigo-executive-smach ros-indigo-smach-viewer \
  ros-indigo-robot-pose-publisher ros-indigo-tf2-web-republisher \
  ros-indigo-move-base-msgs ros-indigo-fake-localization \
  graphviz-dev libgraphviz-dev gv python-scipy liburdfdom-tools \
  ros-indigo-laptop-battery-monitor ros-indigo-ar-track-alvar* \
  ros-indigo-map-server ros-indigo-move-base* \
  ros-indigo-simple-grasping
```

如果你阅读的是本书的纸质版，那复制粘贴可能行不通。不过你可以使用下面的命令来下载一个名为 `rbx2-prereq.sh` 的小的 shell 脚本，脚本将运行上面列出的 `apt-get` 命令。

```
$ cd ~
$ wget https://raw.githubusercontent.com/pirobot/rbx2/indigo-devel/\
  rbx2-prereq.sh
$ sh rbx2-prereq.sh
```

我们还需要本书第一卷（rbx1）代码仓库中的代码（即使你没有那本书）。运行下面的命令来安装 ROS Indigo 版的 rbx1 代码（若你的电脑上还没有安装）：

```
$ cd ~/catkin_ws/src
$ git clone -b indigo-devel https://github.com/pirobot/rbx1.git
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
```

要克隆以及编译第二卷 ROS Indigo 版本的代码仓库（rbx2），执行以下步骤：

```
$ cd ~/catkin_ws/src
$ git clone-b indigo-devel https://github.com/pirobot/rbx2.git
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
```

注意：若你还没有这么做，请把上面的最后一行加到~/. bashrc 文件的末尾。这将保证每次你打开一个新的命令行窗口时，catkin 程序包都被加到 ROS_PACKAGE_PATH 中。

如果 *ROS By Example* 的代码接下来更新了，你可以把更新后的版本和本地保存的代码资源用以下命令合并：

```
$ cd ~/catkin_ws/src/rbx2
$ git pull
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
```

保持更新：如果你想了解更多关于新书和本书中代码的更新情况，请加入ros- by-example 的 Google 群组。

所有的 *ROS By Example* 第二卷程序包都以 rbx2 开头。如果要列出所有程序包，进入 rbx2 元程序包的父文件夹，使用 Linux ls 命令：

```
$ roscd rbx2
$ cd ..
$ ls -F
```

我们应该会看到以下列出的信息：

```
pedestal_pi_no_gripper_moveit_config/   rbx2_bringup/      rbx2_msgs/
pedestal_pi_with_gripper_moveit_config/ rbx2_description/ rbx2_nav/
pi_robot_moveit_config/                 rbx2_diagnostics/ rbx2_tasks/
rbx2/                                   rbx2_dynamixels/  rbx2_utils/
rbx2_arm_nav/                           rbx2_gazebo/      rbx2_vision/
rbx2_ar_tags/                           rbx2_gui/         README.md
```

贯穿本书，我们会使用 roscd 命令来从一个程序包进入另一个程序包。例如，要进入 rbx2_dynamixels 程序包，请使用下面的命令：

```
$ roscd rbx2_dynamixels
```

注意，你在任意目录下运行这个命令，ROS 都会找到这个程序包。

重要：如果你使用两台电脑来控制或监控你的机器人，例如有一个笔记本在机器人上，同时桌面上还有另一台电脑，请确保在两台机器上都复制和构建了 rbx2 和 rbx1 代码仓库的 Indigo 分支。

3 使用 ROS 执行任务

正如我们在第一卷中所发现的，给一个机器人编程以执行如脸部追踪、坐标之间导航或跟踪一个人等特定行为，是相对直接明了的。但是一个全自动机器人会被要求根据当前的任务以及当前的条件，从一个更大的行为集合中选取它自己的动作。

在本章中，我们将学习如何使用两种不同的启用 ROS 的任务执行框架：SMACH（使用了状态机，读作“smash”）和pi_trees（行为树）。这两种方法都有各自的优点和缺点，并且你的编程背景可能决定了你掌握其中一种方法的难易程度。但是，两种方法都提供了一个更为结构化的任务管理途径，而非简单地罗列一长串 if-then 语句。

全局任务控制器一般被称为任务执行器，多数此类执行器都至少包含以下关键特性：

- 任务优先级：若有一项高优先级的任务需要相同的资源（例如：驱动电机），则低优先级的任务应被抢占资源。
- 暂停与继续：当一项高优先级任务被给予了控制权，当前执行的一项（或多项）任务被暂停，而当执行器收回控制权，被抢占的一项（或多项）任务继续进行，这通常是一项可取的特性。例如，Neato 吸尘器就可以在充电之后返回之前中断的任务中。
- 任务层级：任务通常可以拆分成子任务，以进行细节处理。例如，一项被称为再充电的高级别任务可能由三项子任务组成：导航至充电站，将机器人置入充电站以及给电池充电。类似的，将机器人置入充电站的任务又可拆分成：对准指示灯，向前驱动，置入充电站后停止。
- 条件：传感器数据与内部编程变量都可对任务执行的时间与方式产生约束。在 ROS 中，这些变量通常以各种节点发布的消息为形式。例如，电池监控任务会订阅一个带有电池水平的诊断主题。当低电池水平被检测到时，检查电池的条件被激活，再充电的任务将会执行，其他任务则会暂停或中止。
- 并发性：多项任务可以并发进行。例如，导航任务（前往下一个路径点）与电池监控任务必须同时进行。

为了说得具体一些，我们将在这一整章中使用两个案例场景：一个是“巡逻机器人”，它必须在不让电池耗尽的同时有顺序地经过一系列的位置；另一个是“房屋清洁机器人”，它必须到达若干个房间并进行与每间房间相关的清洁任务。