

# 基于 ROS 户外移动机器人软件系统构建

◆黄开宏 杨兴锐 曾志文 卢惠民 郑志强  
(国防科技大学机电工程与自动化学院, 湖南长沙, 410073)

**摘要:** 在机器人功能日益复杂的情况下, 如何简单快速地为机器人构建所需的软件系统是一个值得探讨的问题。本文首先分析了机器人软件系统开发过程中面临的问题并介绍了机器人操作系统 (Robot Operating System, ROS) 这一软件平台的基本概况、基本概念和主要特点, 然后基于 ROS 为本实验室研制的四轮户外移动机器人构建了一套软件系统, 实现了移动机器人的底层驱动控制、远程遥控、识别可通行区域等功能。该软件系统充分展示了利用 ROS 软件平台构建机器人软件系统的灵活性、易用性和功能丰富等特点。

**关键词:** 机器人, ROS, 户外移动, 可通行区域识别

## 0 引言

随着机器人领域的快速发展, 机器人的各项能力在日趋完善, 为机器人编写软件这一工作也变得越来越复杂繁重。

机器人软件系统日益复杂的原因很多, 如:

1) 机器人的功能层次化要求在加深。

机器人越是智能, 软件系统的功能分层越是明显, 从简单的底层硬件驱动到中层的环境感知、运动规划, 再到高层的逻辑推理、行为决策, 最终导致软件系统越来越复杂、庞大。

2) 机器人的计算平台在扩展。

现有机器人的计算平台已经从传统的 PC 平台、嵌入式平台延伸至新兴的智能手机、平板电脑等移动设备, 相应的操作系统也发展到安卓、IOS 等用于移动设备的操作系统。机器人的软件系统是否具备跨平台的能力已经成为开发者考虑的一个重要因素。

3) 机器人用到的硬件越来越复杂。

机器人配备的传感器从简单的光电开关、触碰开关, 发展到超声测距、激光测距, 再发展到如今的激光雷达、RGBD 深度摄像头等复杂设备。这些传感器不断提升了机器人的感知能力, 相伴而来的则有越来越大的数据吞吐量, 这对软件系统的实时性、可靠性等方面提出了挑战。

4) 对软件仿真的需求在加深。

通过计算机对实际的机器人系统进行仿真模拟可以

有效地缩短机器人的开发时间和成本。但是, 开发一套具有实时性和真实性高、通用性强、使用方便的仿真软件往往需要耗费大量的时间。

为了简化编程工作, 许多机器人软件开发平台应运而生。比较著名的商业软件平台有微软公司的 Robotics Developer Studio (MRDS)<sup>[1]</sup> 集成开发环境, NI 公司的 LabVIEW Robotics 模块, LEGO 公司的 MINDSTORMS 套件等等。也有许多来自于高校的开源软件项目, 如美国南加州大学机器人研究实验室开发的 Player/Stage 项目<sup>[2]</sup>、美国卡内基梅隆大学的 CARMEN 工具箱<sup>[3]</sup> 等等。这些软件平台特点各一, 涵盖了机器人领域的各个方面, 如 Player 套件着重强调硬件及驱动的模块化; 也有专门针对仿真功能的软件, 像卡内基梅隆大学的 USARSim<sup>[4]</sup> 和 OpenRAVE<sup>[5]</sup> 仿真环境; 其他的如对一些比较成熟的算法进行封装, 像视觉领域里的 OpenCV 函数库、强调任务规划的 MissionLab 平台等等。文献 [6] 对各种软件平台、仿真环境、工具库做了比较详细的对比。

由于机器人领域的广泛性, 不同的应用环境对机器人的软件系统有不同的要求, 一个平台不可能满足所有应用的要求。但是有一些要求是共性问题, 比如易用性、开发效率、跨平台、多编程语言、分布式部署、代码可重用等等。针对这些问题的解决, 机器人操作系统 (ROS, Robot Operating System)<sup>[7]</sup> 走在了众多软件平台的前列。

ROS 是一个机器人软件平台, 它能为异质计算机集

群提供类似操作系统的功能。它的前身是斯坦福人工智能实验室为了支持 STAIR 和 Personal Robots Program 等大型服务机器人项目而建立的交换庭 (switchyard) 项目。2008 年, Willow Garage 公司完善了该项目的研发, 2009 年正式对外发布 ROS 开源软件项目。

作为一个开源软件, ROS 的宗旨在于构建一个能够整合不同研究成果, 实现算法发布、代码重用的机器人软件平台。它提供类似操作系统所提供的功能, 包含硬件抽象描述、底层驱动程序管理、共用功能的执行、程序间的消息传递、程序发行包管理以及一些用于获取、建立、编写和运行多机整合程序的工具程序和软件库。由于这些功能满足了广大开发者的需求, ROS 得到了广泛的应用。不少研究人员在 ROS 的基础上开发了许多如定位绘图、行动规划、感知、仿真等高层功能软件包, 使得这一软件平台的功能更加丰富, 发展更加迅速。截至目前, ROS 的核心程序已经发行至第六版, 并且以每年两个版本的速度在更新。如今基于 ROS 进行软件开发的机器人已多达 95 种, 其范围涵盖移动机器人、机械手、类人机器人、无人车和无人飞行器等热点领域, 而且大部分传感器在 ROS 上都得到了比较好的支持。基于 ROS 开发的应用程序更是多达 2000 种, 涵盖了硬件驱动、模拟仿真、运动规划、运动控制、环境感知等各个方面。ROS 已经逐步成为机器人研发领域的通用性软件平台。

本文详细介绍 ROS 的主要特点, 并结合实际情况介绍如何利用 ROS 为四轮移动机器人搭建能够实现远程遥控、可通行区域识别等功能的软件系统。

1 ROS 的基本概念和主要特点

与 Windows、Linux 等传统意义上的计算机操作系统不同, 机器人操作系统 ROS 并不负责计算机进程管理与调度。实际上它是一系列运行在上述计算机操作系统上的软件工具集。ROS 的核心功能是提供一种软件点对点通信机制, 基于这个机制, 开发人员可以非常灵活和高效地组织机器人的软件系统。

“节点”、“消息”、“话题”和“服务”是 ROS 中的四个基本概念。

1.1 节点

“节点”是与机器人功能相关的任务进程, 可以认为一个节点就是一个软件模块。这些节点可以是硬件的驱动

程序, 负责采集传感器数据或给执行器发送命令等; 也可以是实现诸如视觉算法、运动规划、逻辑推理等高级功能的运算进程; 或者是帮助调试、记录数据或者监察系统状态的辅助进程等等。整个软件系统可以看作是一个图, 其中任务进程是这个图的节点, 进程之间的点对点通信是连接节点的边。利用 ROS 提供的可视化工具, 开发人员可以实时地观察系统的运行情况, 如图 1 所示, 圆圈代表了不同的节点, 连线和箭头代表了节点间的通信和通信方向。

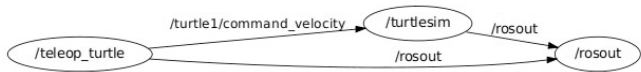


图 1 正在运行的三个节点程序

1.2 消息

“消息”是数据在 ROS 中的表现形式, 是节点之间的通信内容。不管是来自传感器的数据, 还是算法的运算结果, 都可以定义为消息。由于 ROS 是面对多种编程语言的, 为了能够进行跨语言环境开发, ROS 使用了一种名为 IDL 的定义语言来定义消息的结构和类型。利用诸如整型、浮点型、字符串和布尔型等标准的数据类型, 用户可以根据自己需要定义不同的消息, 消息的元素可以是单一变量、数组或者常量。同时, ROS 还支持消息的嵌套, 即一种消息的定义里可以包含其他消息, 而且嵌套深度不受限制。图 2 给出了一个 IDL 消息定义文件的例子。这个名为 Joy.msg 的文本文件定义了一个 Joy 数据消息, 用于记录手柄设备的信息。

Header	header
float32[]	axes
int32[]	buttons

图 2 消息定义文件

IDL 的语法简单明了, 每一行的左边是数据的类型, 右边是数据的名称。图 2 的消息包含了一个 Header 消息、一个 32 位浮点数组和一个 32 位整型数组, 其中 Header 消息也是由类似的方法进行定义。ROS 的代码生成器根据这个 IDL 文本文件自动生成适合目标语言的代码, 并在节点通信的过程中自动对该数据消息进行串行化和反串行化。

以上面数据为例, 代码生成器将自动产生 137 行 C++

代码（或者 96 行 Python 代码、81 行 LISP 代码、99 行 Octave 代码）。因此通过编写简单的 IDL 文本文件就可以实现非常复杂多样的数据消息类型，不但有效减少开发者的编程工作量，还保证了代码的一致性。到目前为止，ROS 平台上已有超过 400 种数据消息，范围涵盖了传感器信息、目标检测信息、地图信息、动作指令等等，基本满足了广大研究人员的需要。

### 1.3 话题

“话题”是传递“消息”的一种形式。每个话题都有一个“名称”字符串作为索引。节点通过订阅某些话题来得到自己感兴趣的数据。这种基于话题发布/订阅的通信形式非常灵活，一个话题可以同时有多个发布者或订阅者，一个节点也可以同时发布或者订阅多个话题，话题的发布者或者订阅者不需要知晓对方的存在，所有发布、订阅行为都是动态进行的。

### 1.4 服务

虽然基于话题发布/订阅的通信形式非常灵活，但是并不适合进行节点间的同步或者事件触发，因此，ROS 引入了“服务”的概念。这与互联网上的服务器—客户端结构类似（如网页服务），作为客户端的节点向提供服务的节点发出请求，接受到请求的节点根据请求中携带的参数进行响应并返回结果。比如在移动机器人中，实现任务规划的“高层”节点可以向负责运动控制的“低层”节点发出移动请求，请求中包括目标点坐标、速度、时间等信息。“低层”节点根据这些信息执行任务，完成后将执行情况返回，比如是否到达了预期的目标点、实际到达点的坐标是多少、运动中消耗了多少能量等等。

### 1.5 点对点通信机制

ROS 的点对点通信方式需要某种查找机制来使得各个进程能够动态地查找到其他进程以便进行通信。在 ROS 里，这个功能由主机（master）实现。主机也被称为命名服务器（name server），与互联网的 DNS 服务器类似，它只提供命名查找服务，并不负责通信数据的中转，这是 ROS 与 CARMEN 等基于数据中心的软件平台的最大区别。

CARMEN 等系统虽然能实现基于网络的分布式运行，但是它们要求其中一台计算机担当数据中心的角色，所有

数据都必须经过这个数据中心进行交换，因而很容易在异构的计算网络中发生问题，比如带宽较窄的无线网络产生拥堵，或者数据中心负荷太重。相反，ROS 采用的点对点通信拓扑结构可以明显减少不必要的流量，因为软件进程可以直接互联进行数据交换，不需要中转。

ROS 的这种通信机制是基于计算机操作系统提供的网络套接字（Socket）实现的，所以 ROS 秉承了网络套接字通信的几个优点：

#### 1) 支持多种编程语言。

由于所有节点的通信都是通过网络套接字实现的，这意味着只要能够提供套接字接口，节点程序可以用任何编程语言实现。ROS 当前已经实现对 C/C++、Python、Octave 和 LISP 等编程语言的支持。

#### 2) 支持分布式计算。

由于大量采用了套接字通信，ROS 系统很自然地具备了实现分布式计算的能力。节点程序可以运行在任何具有网络连接的主机上，这些主机可以是具有强大运算能力的基站服务器，可以是小型的 ARM 嵌入式系统，甚至可以是基于安卓系统的移动手机或者平板电脑。这些异构的运算平台都可以成为机器人系统的一部分。

#### 3) 支持信息资源的任意组合。

#### 4) 支持不同速率、不同数据类型的通信。

## 2 基于 ROS 的户外移动机器人软件系统

移动机器人技术的研究属于多学科相互交叉、相互渗透的前沿课题，对它的研究具有很大的理论价值和广阔的应用前景。本文以实验室研制的四轮车式户外移动机器人为基础，介绍利用 ROS 构建能实现远程操控和环境感知功能的软件系统和基础框架。

### 2.1 机械平台

如图 3 所示，该机器人采用了四轮车式底盘结构，前轮对车体的运动方向进行控制，后轮负责驱动车体前进。前轮的摆动角度由舵机控制，同时配有旋转编码盘对前轮的滚动进行计数，实现里程计的功能。后轮由一个 150W 直流伺服电机驱动，通过配套的电机驱动器实现精确的后轮转速和位置闭环控制。该机器人具有体积小、成本低廉、行进速度快等优点，适合在平坦路面快速前进。目前该机器人最高时速可达 15km/h。



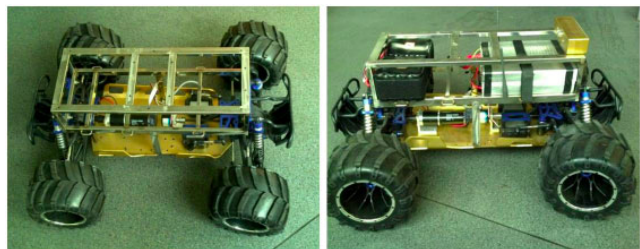


图3 四轮车式户外移动机器人

2.2 硬件系统

机器人的硬件控制系统如图4所示，由监控计算机、机载计算机、嵌入式控制器以及电机驱动器组成。机载计算机采用 Intel Core i7 系列的四核中央处理器，工作频率达 3.5GHz，同时配有 8G 内存以及 64G 固态存储器，可满足机器人对运算性能的需求。嵌入式控制器通过串口与主控计算机相连。该控制器由 ARM 单片机实现，负责将计算机传来的指令转化为对舵机和直流电机的控制信号，并将实时采集得到的测程编码器读数以及电池电压读数送往主控计算机。

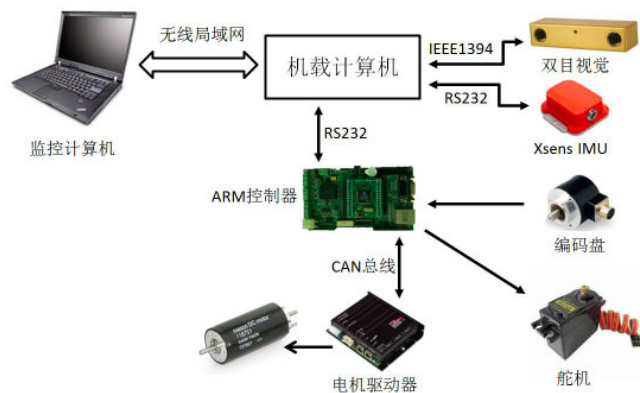


图4 硬件控制系统

在传感器配置方面，该机器人配有 Xsens 惯性测量传感器 (IMU) 以及 GPS 信号接收器，两者配合测程编码器可实现较精确的机器人自定位。该机器人同时配有 bumblebee 双目视觉系统，以实现机器人对户外环境的视觉感知。

2.3 软件系统

利用 ROS 构建的移动机器人软件系统是本文的研究重点。如图5所示，整个软件系统可以根据功能划分为底盘驱动、远程控制、导航模块以及可通行区域识别四个部分。

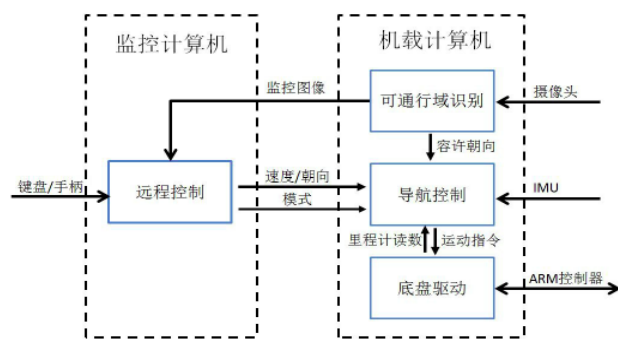


图5 软件系统模块结构

2.3.1 底盘驱动

底盘驱动节点是整个软件系统中最基础的部分，它实现软件与硬件信息互换的功能。在本机器人中，控制车体运动的指令有两部分内容，分别为车体的行进速度与行进方向，对应着直流电机的转速和舵机的摆角。在本软件系统中，控制车体运动的指令以名为 Cmd 的话题进行发布，而底盘驱动节点 (Base 节点) 通常是这一话题的唯一订阅者。Base 节点运行在机载计算机上，它接收到指令以后，根据指令的内容向 RS232 串口发送数据。ARM 控制器再对这些数据进行响应，最终改变机器人的运动状态。

2.3.2 导航控制

导航控制模块是移动机器人的决策核心，其内部建立了一个有限状态机来控制机器人的工作状态。当机器人处于手动控制模式时，该模块从远程控制模块读取操作指令，并以此产生相应的运动指令。当机器人处于自动控制模式时，该模块根据可通信区域识别模块对道路状况的识别结果，控制机器人沿着道路前进。目前自动模式的功能还比较单一，但是可以作为以后开展其他算法研究的基础，比如视觉里程计、同步定位与建图 (SLAM) 算法等等。

2.3.3 远程控制

远程控制是移动机器人的一项基本功能，它涉及指令的产生、传送、执行三个部分。ROS 平台具有实现分布式计算的能力，可以非常容易地实现远程操控功能。在本机器人中，远程控制节点 (Teleop 节点) 是运行在监控计算机上的控制程序。操作人员通过操作监控计算机上的键盘和手柄产生控制指令，Teleop 节点利用 ROS 提供的驱动程序读取到这些设备的输入数据，然后通过发布话题的方式将数据发布到软件节点网络上。机载计算机上运行的 Base 节点通过订阅这一话题来得到指令数据，然后以此

驱动底盘运动。

### 2.3.4 可通行区域识别

ROS 平台得到广泛应用的一个重要原因是它对计算视觉的支持比较好, 不仅对目前主流的摄像头提供驱动支持, 还整合了 OpenCV 这一重量级图像处理软件库。这为开发者节省了大量的时间和工作, 使得开发者可以在较短的时间内搭建出可用的且性能较好的图像处理模块。

对可通行区域进行识别是户外移动机器人实现自主导航的关键环节。针对这一问题, 本户外移动机器人利用 Bumblebee 双目摄像头, 在 ROS 系统提供组件基础上, 实现了准确快速的可通行区域识别功能。

该模块的节点结构如图 6 示, 3 个节点进程呈现类似流水线的结构。其中 Bumblebee2 节点是 ROS 针对该摄像

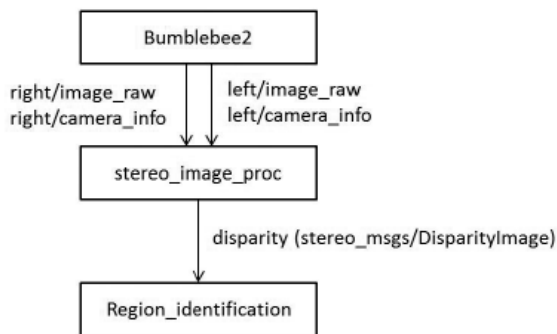


图 6 可通行区域模块的节点结构及话题

头的现成软件包, 它从摄像头读取得到类似图 7 所示的图像数据, 这些图像数据以 left/image\_raw 和 right/image\_raw 话题的形式发布。



图 7 从双目摄像头采集得到的图像

在得到图像数据后, stereo\_image\_proc 软件包可以对图像进行校正、匹配, 最终得到一副 620x320 的深度图像 disparity。最后, Region\_classification 节点利用一种改进的 V 视差图构建算法检测出可通行区域<sup>[8]</sup>, 得到如图 8 所示的结果。实验结果的统计数据表明, 该软件对可通行区域

的识别成功率可达到 92.3%, 满足户外移动机器人自主导航的需要。

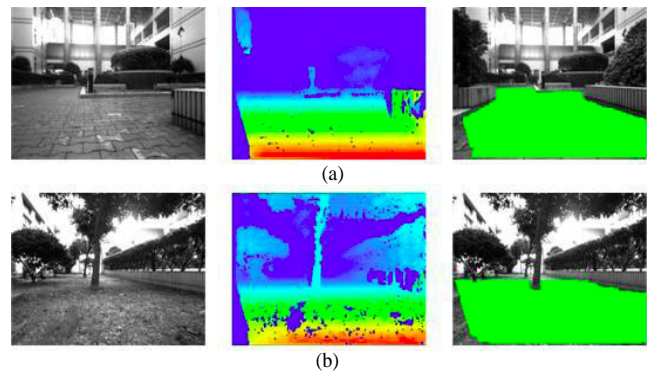


图 8 (左) 原始灰度图; (中) 视差图; (右) 可通行区域结果

该软件模块不仅可以在线地对图像进行实时处理, 还可以利用 ROS 提供的 Rxbag 工具将实验过程中采集得到的图像以及其他通过话题形式发布的数据完整地记录下来, 以便日后进行回放和处理, 有效提高调试机器人的效率。

利用 ROS 提供的开源平台和软件工具集, 我们只需少量的编程工作即可搭建一套较为完整的机器人软件系统。以本文为例, 软件节点共计 11 个, 其中 ROS 提供的节点为 8 个, 占总数量的 72%, 有效地缩短了开发时间, 减少了重复劳动。

由于时间关系, 目前本软件系统实现的功能还比较少, 没有完全利用 ROS 已有的各种功能, 比如 Gazbo 仿真器、同步定位与建图 (SLAM) 工具包、路径规划与导航工具包等等。我们相信, 随着机器人功能的发展, 利用 ROS 构建软件系统的优势将越来越明显。

### 3 结论

本文对 ROS 的优点、原理、特点等方面进行较为详细的介绍, 同时利用基于 ROS 构建的四轮户外移动机器人软件系统, 实现底盘驱动、远程遥控、可通行区域识别等功能, 并充分展示了 ROS 搭建机器人软件系统的灵活性和方便性, 为广大机器人研究人员提供了一套可参考方案。

随着 ROS 这一开源项目的发展, 将会有越来越多的研究人员以及开发者投入到对这一平台的开发与应用中来。我们相信, ROS 未来将为我们提供更多、更复杂的功能, 有效促进机器人领域的交流与发展。(下转 44 页)

的作用，单向阀 12 的开启压力较高，有 0.45MPa，起到回油背压的作用，以控制液压油缸运动的平稳性。

流量控制阀利用负载敏感压力补偿进行优先控制分流，其始终保证将液压泵出口的压力油优先分流给优先口属具快换控制使用，C 口压力油可以控制主机液压系统的其他负载，溢流阀 14 可以限制优先口负载最大工作压力值，节流阀 13 是静态传动，没有产生压力损失，可以提供稳定的负载压力反馈信号，起到补偿器阀芯在移动过程中的稳定性控制作用。

#### 4 结论

通过对属具快换控制液压系统原理的分析，我们不难看出，在属具快换控制液压系统设计选型的时候，要全面考虑快换使用工况需求及其安全保护措施。

此系统在设计时增加了多重安全设计：通过增加液控单向阀，可以锁紧液压油缸无杆腔；通过减压阀可以即时

补油，解决油缸内泄对快换的影响，保证液压油缸无杆腔内压力值恒定；通过安装蓄能器可以在快换或者主机液压系统出故障时，短时间内给液压油缸无杆腔补油，保证液压油缸无杆腔压力值恒定，给配套快换机构的属具工作臂快速下落到安全位置提供时间；通过压力传感器提供低压提前报警功能。

另外，通过选用流量控制阀，可以利用主机上已有的液压泵给快换控制提供液压力源，而不必再增加主机上液压泵的数量。

通过在 20 吨型“双动力双臂手智能型救援工程机械”的一个工作臂上进行工厂内部试验，以及在四川雅安地震救援现场的实际救援作业，均验证了这种属具快换控制液压系统无论在功能原理上还是在安全设计方面，都能很好地满足属具快换的工作需求。🔧

#### 参考文献

[1] HKS TR-K180 用户手册.

(上接 41 页)

#### 参考文献

- [1] Jackson J. Microsoft robotics studio: A technical introduction [J]. IEEE Robotics & Automation Magazine, 2007, 14:82-87.
- [2] Gerkey B, Vaughan R T, Howard A. The player/stage project: Tools for multi-robot and distributed sensor systems//Proceedings of the 11th international conference on advanced robotics 2003[C], 2003:317-323.
- [3] Montemerlo M, Roy N, Thrun S. Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit//Proceedings of the Conference on Intelligent Robots and Systems (IROS) 2003[C].
- [4] Carpin S, Lewis M, Wang J, et al. USARSim: a robot simulator for research and education//IEEE International Conference on Robotics and Automation, 2007[C]:1400-1405.
- [5] Diankov R, Kuffner J. Openrave: A planning architecture for autonomous robotics, CMU-RI-TR-08-34[R]. Pittsburgh: Robotics Institute, 2008.
- [6] Harris A, Conrad J M. Survey of popular robotics simulators, frameworks, and toolkits//2011 Proceedings of IEEE[C]. Southeastcon:243-249.
- [7] Quigley M, Gerkey B, Conley K et al. ROS: an open-source Robot Operating System//ICRA workshop on open source software[C], 2009.
- [8] 朱效洲. 户外环境下基于视觉的移动机器人可通行区域识别研究 [D]. 长沙: 国防科学技术大学, 2012.