**基于 ROS的机器人仿真研究**

侯佳耀

（华南师范大学计算机学院，广东 广州 512000）

**摘要**

本文通过研究基于ROS的机器人仿真，旨在为人型机器人的模拟仿真奠定基础。首先介绍有关机器人相关的历史，奠定研究的历史背景。然后从硬件系统，软件系统和模型建立，仿真环境等方面进行研究探讨。

硬件系统采用嵌入式计算机和底层控制系统组成，控制整个机器人的仿真操作。软件系统采用Ubuntu16版本上运行的ROS kinect。模型搭建与环境的仿真均在这一系统上完成。

通过硬件系统，软件平台，机器人操作系统等多平台联动，最终可以完成人型机器人的相关仿真。

**关键词：**ROS，人型机器人，Gazebo，URDF，Xacro，键盘控制

**abstract**

This paper aims to lay a foundation for the simulation of humanoid robot by studying the robot simulation based on ROS. Firstly, the history of robotics is introduced to establish the historical background of the research. Then the hardware system, software system and model building, simulation environment and other aspects are discussed.

The hardware system is composed of embedded computer and bottom control system to control the simulation operation of the whole robot. The software system uses ROS Kinect running on Ubuntu 16. The model building and environment simulation are completed on this system.

Through the hardware system, software platform, robot operating system and other multi platform linkage, the related simulation of humanoid robot can be completed.

**Key words:** ROS, humanoid robot, gazebo, URDF, xacro, keyboard control

**1.引言**

机器人技术始于上世纪五十年代,德沃尔和英格伯格在 1961 年联手制造了第一台可编程的实用工业机器人 Unimate。随着科学技术的发展,机器人已渗透于人类生活的各个领域。与工业机器人不同,人

形机器人因具有类似人的形态,人们对它的接受度更高。人形机器人已成为当前机器人领域的研究热点 , 其集机械、电子、计算机、材料、传感器、控制技术、 信息、通信、人工智能和人工心理等多门学科于一体,代表了一个国家的科技发展水平 。

几十年来,许多国家陆续推出了不同的人形机器人。同时,随着机器人技术的快速发展,相应的开发环境越来越多、体系也越来越复杂,机器人面临软件通用性不足和复用性差等问题,使其发展面临严峻考验。为提高机器人研发过程中的软件复用率,2010年机器人操作系统(robot operating system,ROS)发布。 ROS 是一个开源机器人操作系统平台,采取分布式结构框架,点对点设计,具有多语言支持、架构简单、集成度高及组件化工具包丰富等特点。

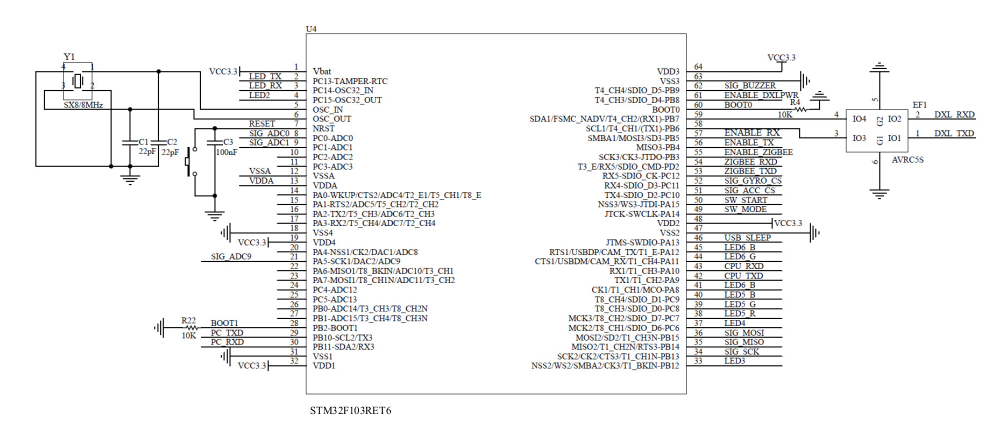
1. **人型机器人**

**2.1硬件系统**

人形机器人硬件系统主要由上层嵌入式计算系统和底层控制系统构成。上层嵌入式计算系统主要由主控制器 PICO-HC101 和 USB 摄像头组成,负责视觉感知和运动规划等复杂算法的计算;底层控制系统主要由子控制器 STM32 舵机控制板、MX-28T 舵机和 IMU 等组成,负责控制舵机和收集 IMU 等传感器信息。

主控制器 PICO-HC101 最高可搭载 2 GHz 的Intel® AtomTM E3845/ Celeron® J1900/N2807 处理器和 8 GB 的 SODIMM,以及各种通信接口包括 USB3.0/2.0, STAT, DP/VGA 和 w/LAN。子控制器 STM32舵机控制板主要负责主控制器、舵机和传感器之间的实时通信,主芯片选用 STM32F103RET6(下文简称 STM32)单片机,并在板上集成了陀螺仪和加速度计传感器。

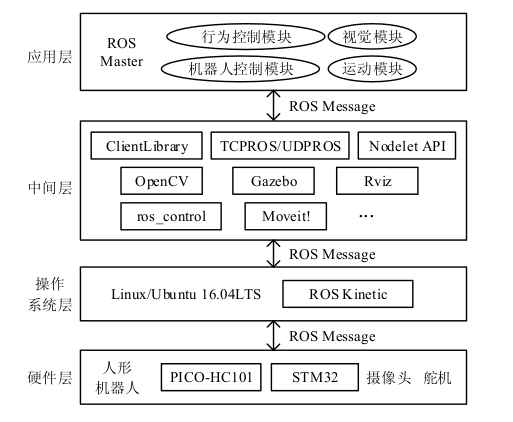
主控制器通过 USB 接口与外部通信,与子控制器的数据收发需要通过 STM32 设计 USB 转串口电路实现;通过 STM32 设计串口转 TTL 及 I 2 C/SPI 实现舵机、IMU 与子控制器的通信。因此,要实现主控制器对机器人的稳定控制,STM32 电路设计尤为重要。STM32 电路图如图 1 所示。



**图1**

**2.2软件系统**

ROS 是一个次级机器人操作系统，点对点的设计，可将机器人的功能模块分割成节点的多个独立进程，每个节点负责处理一个功能模块，且各节点分开编写、编译和启动。不同节点之间通过基于发布/订阅模型的话题和基于客服端/服务器模型的服务，实现消息在节点间的交换。这种连接体系使机器人系统功能实现模块化设计，各模块之间具有较高的独立性。本文设计的基于 ROS 的人形机器人软件框架如图 2 所示，分为硬件层、操作系统层、中间层和应用层。ROS 作为次级操作系统，无法直接在计算机硬件上运行，需要搭载在 Linux 系统上。为此，在主控制器上移植了 Ubuntu 16.04LTS 系统，并选取与之对应的 Kinetic 版本 ROS 作为人形机器人的软件平台。



**图2**

应用层的行为控制模块、机器人控制模块、视觉模块和运动模块等是本软件控制系统的研究重点。将每个功能模块设计成一个节点,并在 ROS Master 管理下正常运行。

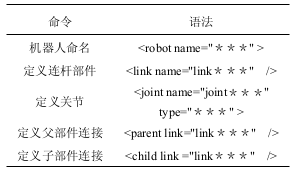
其中,行为控制模块负责较高级别的规划和控制任务,整合处理各模块的话题消息;机器人控制模块是底层控制部分的核心,负责传感器与舵机的实时控制,通过 STM32 舵机控制板和硬件接口组件实现通信,需要设计 2 个硬件接口,1 个用于真实机器人通信,1 个用于仿真的虚拟接口模拟通信。

视觉模块是机器人感知周围环境的重要功能模块,可以细分为视频采集、目标检测跟踪和目标测距等多个子节点;运动模块具有步态、头部控制、动作控制和跌倒检测 4 个核心子节点。

1. **模型建立**

**3.1 URDF 文件**

URDF 是 ROS 中使用 XML(可扩展标记语言)格式描述机器人模型的文件 ,包括机器人的外观形状、尺寸、颜色、物理属性和关节类型等基本属性。编写 URDF 文件需要遵守的基本编程语法（如图3）。其中,<link>标签描述机器人各连杆部件;<joint>标签描述机器人各关节;<parent>和<child>描述关节连接的连杆部件; <robot>标签是整个机器人模型的顶层标签,包含其他所有标签,一系列的<link>和<joint>组合起来就是一个完整机器人框架。

****

**图3**

**3.2 URDF的模型构建**

为实现基于 ROS 的人形机器人 URDF 文件描述,需要对其<link>和<joint>进行定义。由于 URDF 中的标签只提供 box, cylinder 和 sphere 等基本形态,因此可使用<mesh>标签导入 COLLADA (.dae)或 STL(.stl)格式的外部文件。

首先,使用图3中的基本指令构建不考虑模型尺寸大小信息的 URDF 基本树形结构,并使用<mesh>标签导入三维模型;然后,使用 check\_urdf 命令检查所构建的 URDF 是否存在语法错误,如无错误,在终端中输出成功的信息。

**3.3 展示URDF模型**

Rviz 是 ROS 根据机器人系统可视化需求提供的一款三维可视化工具,可以显示 URDF 文件描述的机器人模型、运动状态、传感器信息和周围环境等多种数据。

实现 Rviz 对机器人的显示,需要在 launch 文件中配置参数加载机器人 URDF 模型;配置 joint\_state\_publisher 节点以发布机器人关节状态;配置 robot\_state\_publisher 节点以发布坐标变换关系;配置 Rviz 节点以运行 Rviz 可视化界面。

**3.4 Xacro与URDF的结合**

XACRO文件和URDF实质上是等价的。XACRO格式提供了一些更高级的方式来组织编辑机器人描述。主要提供了三种方式来使得整个描述文件变得简单。借用一句话来形容xacro的优势: “Fortunately, you can use the [xacro](http://wiki.ros.org/xacro) package to make your life simpler”。

Xacro在定义上可以广泛使用宏定义和宏引用，这使得Xacro与Xacro，Xacro与URDF可以紧密联系起来，形成宏体系。这也是Xacro对比单一使用URDF进行模型构建的绝对好处之一。

1. **仿真环境**
   1. **Gazebo**

Gazebo 是一个三维物理仿真平台,因具有强大的物理引擎及高质量的图形渲染,使其可以提供高保真度的物理模拟,从而得以在复杂环境中准确、有效地模仿机器人。因此,在不具备机器人实体的情况下,通过 Gazebo 来仿真机器人是一个较好选择。搭建 Gazebo 物理仿真环境,首先,为人形机器人配置 Gazebo 属性,分为以下 4 步:

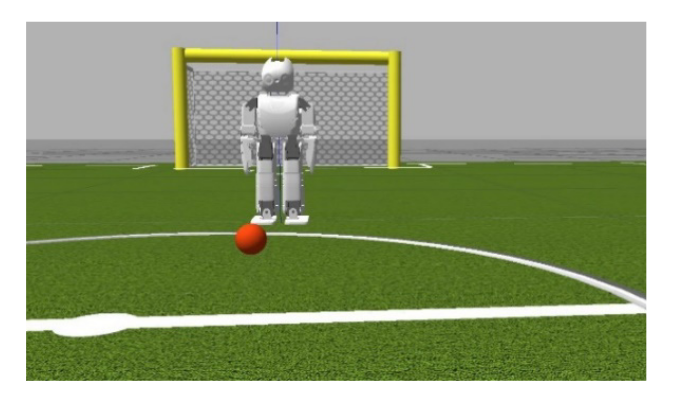
1) 为 link 添加惯性参数<inertial>和碰撞属性<collision>

2 ) 为 link 添 加 <gazebo> 标 签 , 并 设 置 其<material>属性

3)添加传动装置<transmission>

4)添加 Gazebo 控制器插件

其次,配置 launch 文件,将人形机器人模型加载到 Gazebo 仿真环境中;最后,直接添加环境模型或使用 Building Editor 创建仿真环境。人形机器人在Gazebo 中的仿真环境如图 4所示。

****

**图4**

* 1. **Gazebo运动控制**

Gazebo是一个大型的仿真环境，但是其缺乏控制功能，故加入了相应的控制节点。例如键盘控制节点mrobot\_teleop，通过这一节点可以在终端中使用键盘控制机器人在仿真环境中移动并参与扫描SLAM等高级工作。使用键盘控制节点的步骤分为以下几步：

1）使用模型展示功能使机器人在rviz中同步显示

2）启动键盘控制节点使机器人拥有键盘控制功能，其指令为：

roslaunch mrobot\_teleop mrobot\_teleop.launch

1. **结语**

本文在人形机器人硬件设计基础上,介绍了 ROS平台中使用 URDF 文件构建机器人模型的方法,实现了人形机器人外壳的机器人三维模型构建,并搭建了相关物理仿真环境。通过Gazebo 进行相关控制器的参数配置,对人形机器人进行了初步仿真。仿真结果表明:本文设计的基于 ROS 的人形机器人可进行有效运动, 验证了该模型的合理性,并为人形机器人在 ROS 环境中进行复杂的运动规划及控制打下了理论基础。

**参考文献**

[1]袁丹鹤;杜玉晓;江鑫;向颖;王焕.基于ROS的人形机器人建模与仿真[J].自动化与信息工程,2020,v.41,9-15.

# [2]小贝也沉默，<ROS> 机器人描述--URDF和XACRO，CSDN

[3]冯刚，机器人slam与自主导航PPT