

Design of Semi-Virtual Manipulator Based on ROS

Xinglu Ma, Jijun Qin

College of Information Science & Technology, Qingdao University of Science & Technology, Qingdao Shandong
Email: 992132844@qq.com

Received: May 4th, 2020; accepted: May 18th, 2020; published: May 25th, 2020

Abstract

Virtual simulation teaching system has great advantages in principle explanation and confirmatory experiment, but it also has some disadvantages, for example, real state cannot be simulated, students lack real exercise, and so on. For this reason, this paper proposes a semi-virtual design idea, which focuses on part of the experiment and peripheral auxiliary equipment, and designs a semi-virtual mechanical arm teaching system, which is used in the teaching of embedded system courses of computer specialty. Based on ROS and using gazebo as simulation software, the virtual simulation scene and virtual interface card running on PC are designed in this design. Together with the real embedded system development device, it forms a complex semi-virtual training scene, which enables students to gain development experience on the real development device, and verify the software function by using the virtual simulation scene. This method has both the advantages of full virtualization and real equipment, and has a greater advantage in the cultivation of students' ability.

Keywords

Semi-Virtualization, Robot Arm, ROS, GAZEBO

一种半虚拟化机械臂教学系统的设计

马兴录, 秦吉军

青岛科技大学信息科学技术学院, 山东 青岛
Email: 992132844@qq.com

收稿日期: 2020年5月4日; 录用日期: 2020年5月18日; 发布日期: 2020年5月25日

摘要

虚拟仿真教学系统在原理讲解及验证性实验方面具有较大优势,但也存在一些真实状态无法模拟、学生

缺乏真实锻炼等缺点。为此, 本文提出将实验关注部分真实化、外围辅助设备虚拟化的半虚拟化设计思路, 设计了一种半虚拟化机械臂教学系统, 用于计算机类专业嵌入式系统课程的教学。本设计基于ROS, 采用GAZEBO作为仿真软件, 设计了运行于PC机的虚拟仿真场景、虚拟接口卡, 与真实的嵌入式系统开发装置共同构成一个复杂的半虚拟化实训场景, 让学生在真实的开发装置上获取开发经验, 利用虚拟的仿真场景验证所设计的软件功能。该方式兼具全虚拟化及真实设备两个方面的优点, 对学生能力培养具有较大优势。

关键词

半虚拟化, 机械臂, ROS, GAZEBO

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

目前机械臂作为比较成熟的机器人, 很多大学都采购了各种机械臂作为实验教学设备。但是应用于实验教学中的实体机械臂往往会受到设备数量的限制, 并且设备极容易受到损害。所以可以利用仿真技术, 通过交互式计算机图形技术和机器人学理论等, 在计算机中生成机械臂的三维几何图形并显示, 用来确定机械臂的本体及工作环境的动态变化过程, 通过系统仿真可以在操作实体机械臂之前验证操作算法及控制程序的准确性, 避免直接对设备操作以减少对设备的损害[1]。但目前高校中对机械臂的仿真实验是在PC机上完全虚拟化操作, 学生只能给仿真软件修改参数以及编辑代码运行查看仿真结果, 缺乏对实体机械臂控制的真实性, 难以培养学生解决实际问题的能力。

基于此, 设计一种基于ROS操作系统的半虚拟化仿真, 通过设计的一个硬件接口板与显示虚拟化仿真结果的PC设备相连接, 能够让学生在真实的开发环境中操作, 并在虚拟化的设备上显示操作结果, 即可解决实验设备紧缺、损坏的局限性又可培养学生真实的开发环境中解决实际开发问题的能力。

2. 半虚拟化仿真系统设计方案

2.1. 仿真设计结构组成

本设计的主要思路是: 应用场景虚拟化, 学生训练用的开发环境是真实的, 虚拟化场景与真实开发环境之间采用专门设计的接口卡进行连接, 如图1所示。其中, 部件1为本设计的虚拟仿真软件, GAZEBO动力学仿真器, 它是包含在ROS系统中的一个标准功能包集(Stack), 可对机器人进行物理学特性仿真, 它集成了目前常用的各类传感器, 如激光测距传感器、RGB图像传感器、深度点云传感器、重力加速度传感器等[2]。部件3为专门设计的虚拟接口卡, 作为将嵌入式设备发送的指令送给仿真软件, 进行信号传输的中间件。部件5为嵌入式系统开发板, 可为任意的单片机设备, 如: 51单片机。部件2、3为通讯连线。

其中, 虚拟仿真场景基于ROS操作系统进行设计, 仿真机械臂的运动。ROS操作系统分为三个层级: 组织ROS文件结构的文件系统级(Filesystem level)、管理进程及通信的计算图级(Computation Graph level)和包括社区开发人员及其算法代码的开源社区级(Community level) [3]。其中计算图级是最重要的概念, 主要包括节点管理器(Master)、参数服务器(Parameter Server)、节点(Node)、消息(message)、服务(service) [4]。

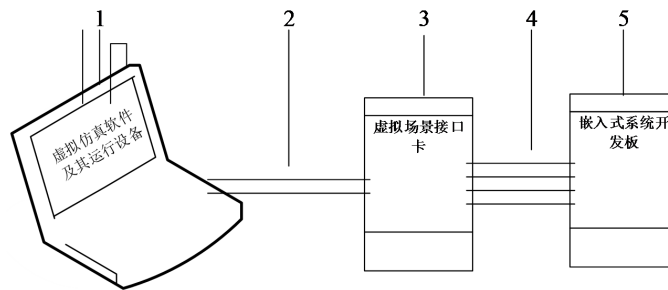


Figure 1. Composition diagram of semi virtualization simulation design
图 1. 半虚拟化仿真设计组成图

ROS 的研发是为了在机器人的研究中可以复用已有内容, 从而提高工作效率。它采用了一种解耦合架构, 使得各个程序可以分别设计, 并进行有机结合, 按照使用的各种包的形式分别管理[5]在 ROS 中有很多 GAZEBO 的插件, 两者之间可以通过话题通信。使用 GAZEBO 也可以与 ROS 中的 Rviz 三维可视化工具联系起来, 将得到的数据直接在 Rviz 中显示出来, 方便开发人员搭建需要的仿真环境。通过仿真环境的测试, 减少开发成本, 加快开发速率[6]。

2.2. 仿真设计操作流程

在本设计中, 首先在 ROS 中基于 GAZEBO 创建好三轴机械臂模型用于仿真。然后将仿真软件的运行设备(PC 端)与专门设计的虚拟接口卡相连接, 再将虚拟接口卡连接至嵌入式开发环境。由嵌入式开发设备发送控制命令, 传送到虚拟接口卡, 虚拟接口卡将控制命令转成脉冲信号发送到 PC 端并由 PC 端解析串口信号, 最后通过 ROS 进行通信, 控制仿真软件中机械臂运动, 操作流程如图 2 所示。

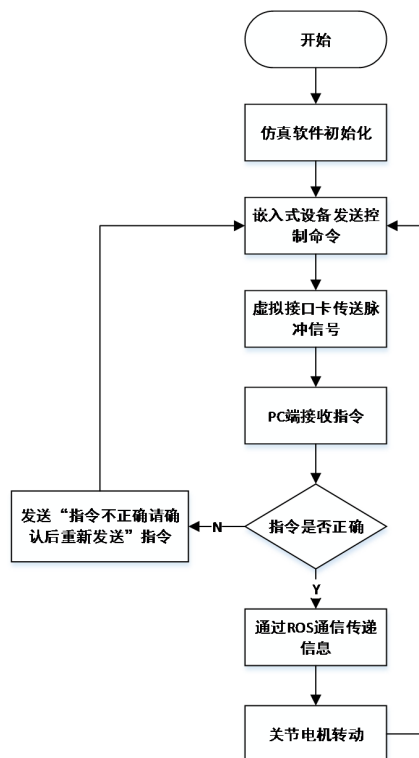


Figure 2. Design operation flow chart
图 2. 设计操作流程

3. 机械臂仿真模型搭建

3.1. 机械臂建模

在 ROS 中通常使用 URDF 来描述机器人的结构、外观以及重量等基本属性, 文件主要由 LINK 和 JOINT 标签组成[7]。其中 LINK 标签描述的内容有机器人连杆的几何、运动以及动力学等特性, 而 JOINT 标签则用于机器人关节及关节与连杆之间的相关特性的描述。这两个标签内都有另外的子标签用于描述机器人的几何信息(颜色、位置等)以及物理信息(重量、惯性动能等)。

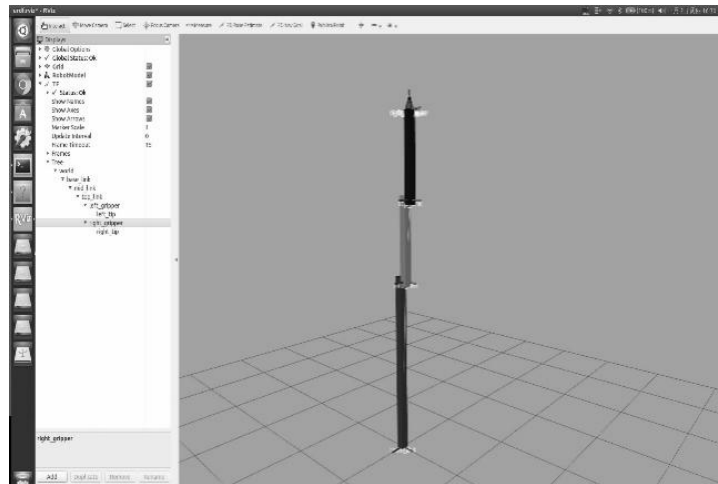


Figure 3. Display manipulator modeling in Rviz

图 3. Rviz 中显示机械臂建模

根据以上对标签的介绍, 可以在 ROS 中创建一个三连接杆、三关节组机械臂模型的 URDF 文件, 该机械臂的启动文件位于 `ros_robotics` 文件夹中。打开一个新终端, 输入命令“`roslaunch ros_robotics rrbot_rviz.launch`”, 可在可视化工具 RVIZ 进行显示, 如图 3 所示。

但是当前的模型只是具备运动学和动力学属性, 在 GAZEBO 中不能正常运行, 所以接下来进行 GAZEBO 仿真的配置。

GAZEBO 仿真软件中机械臂模型文件为 SDF 格式, 此格式与 URDF 格式十分相似, 例如描述机械臂内容的 XML 标签, 很多与 URDF 文件相同。通过在 URDF 文件中添加 `<gazebo>` 标签, 在 GAZEBO 仿真时可以将 URDF 文件自动转换成可识别的 SDF 格式的机械臂描述。再添加 `<material>`、`<mu1>`/`<mu2>` 标签以在 GAZEBO 中能够显示机械臂每个连杆的颜色以及摩擦系数。此外 GAZEBO 中所模拟的物体具备现实世界物体的物理学特性, 因此在其内构建的机械臂会由于重力的影响, 不会一直直立, 终会倒塌, 所以需要机械臂模型设定在 GAZEBO 的基础坐标 `world` 下, 在机械臂模型文件中添加相应配置如图 4 所示:

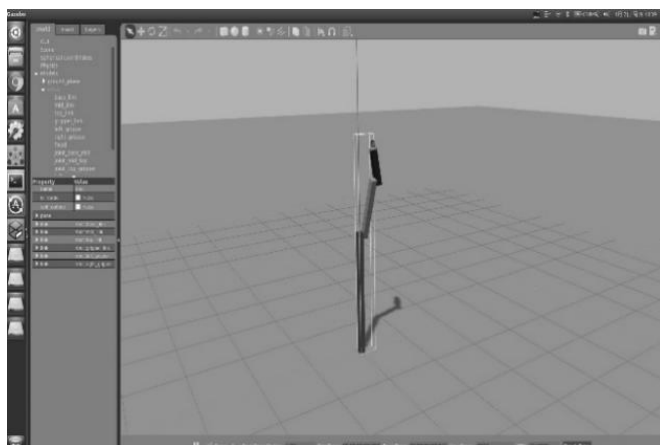
```
<!-- Used for fixing rrbot frame to Gazebo world frame -->
<link name="world"/>

<joint name="fixed" type="fixed">
  <parent link="world"/>
  <child link="base_link"/>
</joint>
```

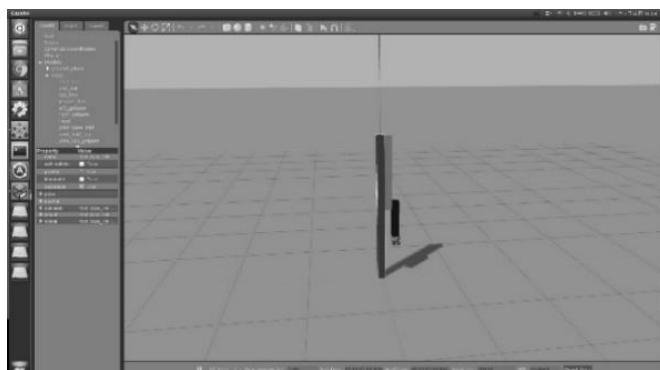
Figure 4. The mechanical arm model is fixed under the basic coordinate world

图 4. 机械臂模型固定在基础坐标 world 下

完成以上, 在新的终端输入命令“`roslaunch ros_robotics rrobot_gazebo.launch`”, 就可在 GAZEBO 中显示虚拟机械臂模型了, 如图 5(a)和图 5(b)。



(a)



(b)

Figure 5. (a) Initial upright state of manipulator in gazebo; (b) Mechanical arm collapsed in gazebo

图 5. (a) 机械臂在 GAZEBO 中初始直立状态; (b) 机械臂在 GAZEBO 中倒塌状态

3.2. 仿真机械臂实现 ROS 通信

当前配置可在 GAZEBO 中仿真显示机械臂模型, 但要让 GAZEBO 中机械臂能够通过 ROS 通信, 还需添加额外控件。

首先为控制在 GAZEBO 中的机械臂模型, 需要在 URDF 文件中对于机械臂的每个旋转副关节添加 `<transmission>` 标签以定义机械臂关节和执行器的相互关系。

然后需定义一个关键插件: `gazebo_ros_control`, 用来解析上述添加的 `<transmission>` 标签并加载相应的硬件接口以及控制器管理器。此外还需要进行关节控制器参数的设置, 创建 YAML 文件, 在其内定义 `joint_state_controller` 控制器以及本仿真机械臂模型的所有关节控制器的控制器类型, 并且在每个关节控制器内还需定义控制器的比例积分微分增益。

最后创建启动文件 `rrbot_control.launch`, 通过 `<rosparam>` 标签在机械臂初始化时可将上述 YAML 文件中所有参数加载到参数服务器中, 再配置 `control_spawner` 节点和 `robot_state_publisher` 节点。在新的终端输入命令“`roslaunch ros_robotics rrobot_control.launch`”即可启动所有的 `ros_control` 控制器。这样就完

成了 GAZEBO 中机械臂与 ROS 通信的联系。

4. 半虚拟化仿真实现

4.1. 虚拟接口卡设计

虚拟接口卡是连接真实开发环境与虚拟设备之间的信息通道，它负责两方面的信息传输：一方面是开发板发出的控制信号，用于控制虚拟设备的动作，对电脑仿真软件来说，成为输入信息；另一方面，虚拟设备的传感器、开关量等信息需要通过该接口板送给开发环境，成为输出信息。因此，虚拟接口卡要及时准确地将双方信息传递过去。

虚拟接口卡有两种实现方式：基于嵌入式微处理器的低速接口卡；基于 CPLD/FPGA 的高速接口卡 [8]。基于嵌入式微处理器的虚拟场景接口板电气原理图如图 6 所示。

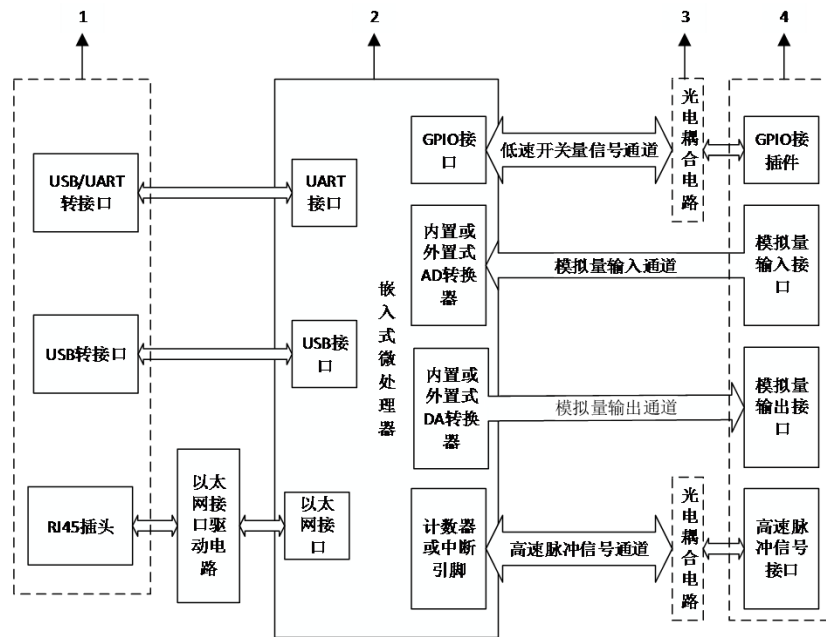


Figure 6. Electrical schematic diagram of virtual scene interface board
图 6. 虚拟场景接口板电气原理图

由图 6 可以看到，编号 2 部分是嵌入式微处理器，单片机或者 ARM 处理器都可以。根据使用场景不同，模拟量输入通道及输出通道、转换器是可以改变的。如果虚拟应用场景中只有数字量信号，则可选用不带 AD 转换器和 DA 转换器的微处理器。针对高速数字量信号，如脉冲信号，则使用嵌入式微处理器的计数器或中断引脚进行捕捉或输出 [9]。

编号 1 部分即为虚拟接口卡与运行虚拟仿真软件的 PC 端的通讯接口，图中所示的三种接口可任意选择。

编号 3 部分的目的是提高接口板的可靠性。

编号 4 部分包含了各种接口信号，如 GPIO、输入输出等。

4.1.1. 输入信息处理

按信息种类来分，输入信息可分为开关量和模拟量两类，对于开关量信息来说，按连接方式分，又可分为总线连接方式和非总线方式两类。对于非总线连接方式，当传送给虚拟接口卡任一开关量输入信

号发生改变, 该接口卡都要检测到并及时传送给电脑。开关量信号变化的检测可有如下三种方案:

1) 定期扫描检测方式: 接口板核心电路采用单片机组成。单片机不断读取输入引脚的状态, 与上次保存的状态比较。如果不同, 表示有引脚的电平发生了变化, 可以立即上报给电脑。

2) 边沿触发中断方式: 任一开关量信号发生变化, 检测其边沿的变化, 并连接至单片机的中断引脚。在单片机中断处理程序中, 直接读入各输入引脚状态, 并立即上报给电脑即可。但该方式对于多路信号的同时检测比较困难, 而且速度与方式 1 相比没有大幅度的提高。

3) CPLD 或 FPGA 的实现: 采用 CPLD 或 FPGA 来检测信号变化, 响应速度只取决于内部逻辑器件的级联门数及门级之间的传输延迟, 速度可在 ns 级。但与电脑之间的传输方面实现比较复杂, 需要实现 UART 接口或 USB 接口, 实现成本较高。

由于虚拟机械臂的控制采用步进电机进行驱动, 采用数字脉冲信号, 因此, 虚拟接口卡采用方式 1 是比较合适的。本设计采用 STC15F2K60S 单片机作为接口卡的处理器。该单片机基于 8051 内核, 可以在每个时钟脉冲运行 1 条指令, 在 12 MHz 晶振频率下, 就可达到 1/12 us 运行一条指令, 可以捕捉微秒级的脉冲信号, 基本满足步进电机控制信号的变化频率。

4.1.2. 接口卡与电脑之间通讯

1) 通讯速率

不管是输入或者输出信息都需要通过通讯方式进行传输, 当输入信号的扫描周期达到 20 KHz 时, 要一次传输 32 路开关量信息, 采用串行传输方式, 每传送一个字节需要增加起始位和停止位, 因此通讯速率需要达到 $20 \text{ KHz} \times (32 + 8) \approx 800 \text{ Kbps}$ 。本设计采用 USB 接口通讯方式完全可满足要求。

2) 通讯协议

尽量缩短通讯协议包的长度, 可以加快传输速度。协议包格式如下:

协议包分为 4 字节格式和单字节格式两类。在 4 字节格式中, 首字节的最高位为 1, 其他字节的最高位为 0; 首字节的次高位为 0, 表示 4 字节格式, 该位为 1, 表示单字节格式。4 字节格式如图 7 所示。

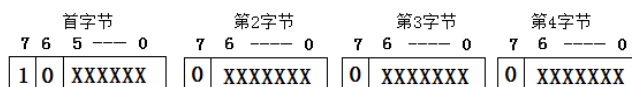


Figure 7. 4-byte communication format

图 7. 4 字节通讯格式

单字节格式如图 8 所示:

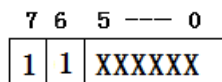


Figure 8. Single byte communication format

图 8. 单字节通讯格式

X 可用于表示 1 个开关量输入信号。这样, 4 字节格式可传输 27 位开关量输入信号; 单字节仅用于传输 6 位信号。当输入信号较少, 或者只有最低 6 为输入信号发生变化时, 使用单字节格式进行传输可提高传输速度。

举例: 如果控制 1 个步进电机, 只需要方向信号 DIR(bit1)和脉冲信号 PULSE(bit 0)即可。这样, 接口板检测到这两个信号中的任何一个发生变化, 都会用单字节协议上传。如图 9 所示, 在 T1 时刻, 接口板会上报电脑单字节协议 0xa2, T2 时刻上传 0xa3, T3 时刻上传 0xa2,T6 时刻上传 0xa1,。

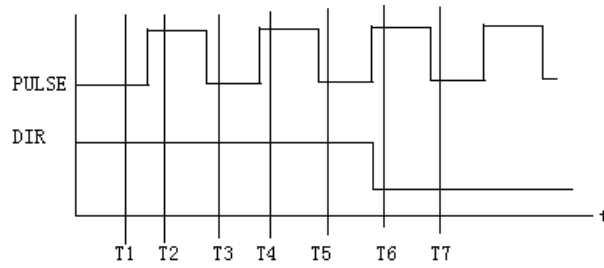


Figure 9. Sequence diagram of step motor control signal
图 9. 步进电机控制信号时序图

4.2. 半虚拟化仿真程序设计

将嵌入式开发板、虚拟接口卡、仿真软件运行设备(PC 端)连线接通, 嵌入式开发板发送控制信号, 虚拟接口卡接收后通过串口给 PC 机发送控制命令, 然后通过 ROS 控制仿真机械臂运动并在 GAZEBO 中显示。在 ROS 中对机械臂进行运动规划是通过 Moveit!软件包来控制。Moveit!是 ROS 中的一个主程序库, 在研究机械臂的多个领域中, 如运动规划、碰撞检测抓取放置等, 都可以应用 Moveit!来解决问题[10]。

启动 GAZEBO 仿真软件显示机械臂模型并启动 ROS 的控制器后, 打开另一终端, 在 ROS 中启动基于 PYTHON 的串口识别程序, 如图 10 所示。在嵌入式开发板上发送一个控制信号, 由虚拟接口卡接收后通过串口发送控制命令给 PC 端, 在 PC 端中的 PYTHON 文件中解析出所有控制命令。例如在 PC 端接收到虚拟接口卡的 P1.0 指针发送了高电平, P0.0 指针发送的低电平, 根据已经定义好的虚拟接口卡指针与机械臂关节电机对应关系, 该指令是让机械臂的一号关节按顺时针转动 9 度。然后给机械臂关节转动弧度参数赋值 0.157 弧度, 并将此参数传给机械臂运动控制程序, 可在 GAZEBO 仿真软件中查看机械臂运动。部分关节电机转动参数设置程序如图 11 所示。

```
# 获取第一个串口名字
port_serial = list(port_list)[0][0]
# 使用USB连接串行口, 获取串口对象
ser = serial.Serial(port_serial, 115200, timeout=520)

while True:
    # 如果接收的缓冲中有字节则读取
    while ser.inWaiting() > 0:
        # 将串口数据传递给解析串口指令方法, 做下一步处理
        parsing_serial_instructions(ser.read_all())
```

Figure 10. Serial port identification program
图 10. 串口识别程序

```
"""
遍历解析好的串口数据列表
列表中每个对象的第一个元素为1或0, 代表高/低电平; 第二个元素为1或-1, 代表电机顺/逆时针运动。
将高/低电平与step相乘得到电机脉冲数值, 将每个对象第二个元素与dir相乘得到电机方向数值,
最后将电机脉冲数值与方向数值相乘得到关节电机最终运动数值。
将每个关节电机的最终运动数值>(*step/-step/θ)存入关节电机运动参数列表中。
"""
for joint_parameters in joint_parameters_list:
    step = joint_parameters[0] * step
    dir = joint_parameters[1] * dir
    joint_rotation_list.append(step * dir)

# 将关节电机运动参数列表传入机械臂关节运动程序中, 控制机械臂运动
MoveRobotArm().joint_rotation(joint_rotation_list)
```

Figure 11. Shutdown motor rotation parameter setting procedure
图 11. 关机电机转动参数设置程序

5. 结论

该系统的半虚拟化设计, 在 PC 端中的 GAZEBO 仿真软件中显示虚拟机械臂作为虚拟操作场景, 再将虚拟接口卡和嵌入式系统开发板、PC 端即电脑相连, 实现了由外部硬件设备发送控制指令控制虚拟仿真机械臂运动, 提供了一种可改善传统本科教学实验中机械臂设备成本高、损坏率高和全虚拟化仿真实验真实性较低的解决方案。既可满足对机械臂运动控制的教学实验, 又可有真实的控制体验。并且虚拟化所具有的封装特性, 能使得控制程序可以重用、共享并且不受地点的限制, 可随时随地在宿舍或者教室里操作。这样的设计, 既提高了学生学习兴趣和效率而且还可培养实际开发能力。

基金项目

2017 年山东省重点研发项目“基于多肽的抗污染肿瘤标志物检测仪的研制”(2017GSF218088)。

2018 年山东省本科教改项目“多学科交叉融合的软件工程专业创新型人才培养模式研究与实践”(M2018X130)。

参考文献

- [1] 邱江艳. 遥机器人人机交互系统及运动规划算法研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨理工大学, 2010.
- [2] 陈君. 一种刚软混杂机器人系统的耦合仿真方法[D]: [硕士学位论文]. 深圳: 中国科学院大学(中国科学院深圳先进技术研究院), 2019.
- [3] 祝传蜜. 基于 ROS 的棚室采摘机器人控制系统平台研究[D]: [硕士学位论文]. 武汉: 湖北工业大学, 2018.
- [4] 藏刘龙. 基于深度学习的移动机器人避障研究[D]: [硕士学位论文]. 长春: 长春理工大学, 2018.
- [5] 姜玉, 原明亭. 基于 Gazebo 的机器人灵活操作控制平台的搭建[J]. 工业控制计算机, 2018, 31(12):44-46.
- [6] Sokolov, M., Gabdullin, A., Afanasyev, I., *et al.* (2016) 3D Modelling and Simulation of a Crawler Robot in ROS/Gazebo. *International Conference on Control, Mechatronics and Automation*, December 2016, 61-65.
<https://doi.org/10.1145/3029610.3029641>
- [7] 梁洪濯. 一种基于 ROS 的新型家庭服务机器人设计与实现[D]: [硕士学位论文]. 马鞍山: 安徽工业大学, 2016.
- [8] Zhu, M. (2010) *The Next Hundreds Embedded Real-Time Operating System*. CoreTek Systems, Inc.
- [9] Sandmann, T., Richter, A., Heyszl, J. and Lübbers, E. (2017) Hardware/Software Trade-Offs for Shared Resources Virtualization in Mixed-Criticality Automotive Multicore Systems. *IT-Information Technology*, **59**, 223-231.
<https://doi.org/10.1515/itit-2017-0018>
- [10] 刘一鸣, 许辉, 耿长兴, 陈国栋. ROS/Gazebo 环境下的机械臂运动规划研究[J]. 煤矿机械, 2018, 39(3): 42-44.