

基于 ROS 的协作机器人控制系统^①



邹 洵, 张 帆, 张国胜, 马保平, 张召颖

(上海工程技术大学 机械与汽车工程学院, 上海 200072)

通讯作者: 张 帆, E-mail: pdszhangfan@sues.edu.cn

摘 要: 为了实现协作机器人的控制, 对协作机器人进行了控制系统研究, 在保证系统鲁棒性和实时性的前提下, 提出了在 PC 机上, 构建基于 Ubuntu 系统下结合 ROS, 利用 CAN 通讯的方式, 搭建协作机器人的控制系统. 最后通过仿真实验和实体机器人控制实验, 验证协作机器人控制系统应用效果. 实验结果表明, 该协作机器人控制系统具备控制协作机器人进行路径规划的基本工作能力, 能够很好的建立上下位机的通讯, 进行协作机器人控制. 同时本控制系统具有模块化、可移植性高、框架清晰、低延时等特点.

关键词: 协作机器人; ROS 系统; 控制系统设计; 通信接口

引用格式: 邹洵, 张帆, 张国胜, 马保平, 张召颖. 基于 ROS 的协作机器人控制系统. 计算机系统应用, 2019, 28(7): 96-100. <http://www.c-s-a.org.cn/1003-3254/6989.html>

ROS-Based Collaborative Robot Control System

ZOU Xun, ZHANG Fan, ZHANG Guo-Sheng, MA Bao-Ping, ZHANG Zhao-Ying

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 200072, China)

Abstract: In order to realize the control of the cooperative robot, the control system of the cooperative robot was studied. On the premise of ensuring the robustness and real-time of the system, it is proposed to build a control system based on Ubuntu system combined with Robot Operating System (ROS), and use Controller Area Network (CAN) communication to build a collaborative robot. Finally, through the simulation experiment and the physical robot control experiment, the application effect of the collaborative robot control system is verified. The experimental results show that the cooperative robot control system has the basic working ability to control the coordinated robot for path planning, and can well establish the communication between the upper and lower machines and control the cooperative robot. At the same time, the control system has the characteristics of modularity, high portability, clear frame, and low delay.

Key words: collaborative robot; ROS system; control system design; communication interface

当前, 随着对机器人应用的大力普及与精度要求的不断提高, 其硬件设计和控制系统研发层出不穷. 由于机器人型号日益增多, 故而需要研发各种控制系统^[1].

目前, 虽然机器人研发公司众多, 但是他们研发的控制系统都是对单一机器人本体做针对性的研发, 普遍适用性和市场推广度较低, 不利于机器人控制系统

的发展. 因此, 需要开发出一套拥有普遍适用性, 契合当下大多数主流机器人的控制系统才能顺应当前市场的刚需.

在实现与主流机器人相配套的控制系统设计, 可充分考虑利用一些开发平台, 如: Microsoft visual, Matlab, MRDS 等. 这些平台各有特点, Microsoft

① 基金项目: 上海市科委生物医药领域科技支撑计划 (17441901200)

Foundation item: Sci-Tech Supporting Plan on Biomedical Field of Shanghai Municipal Science and Technology Commission (17441901200)

收稿时间: 2019-01-18; 修改时间: 2019-02-19; 采用时间: 2019-02-25; csa 在线出版时间: 2019-07-01

visual^[2]适合在 windows 系统环境下开发,较为全面,但过程繁琐,程序易报错; Matlab^[3]多用于仿真实验,但难以实现实体机器人的控制工作; MRDS^[4]重点在图形化编程,但应用范围较小. 并且由于这些软件对于部分算法做了封装,如: OpenCV, Oimpl 等. 所以需要寻求一个能够兼容各种算法,实现多平台运用,上手容易的开发平台.

为此,本文基于开源机器人操作系统 ROS (Robot Operating System),搭建了控制系统平台^[5],进行了控制系统设计并作出了相应的仿真及实物控制实验.

1 ROS 配置环境的搭建

Robot Operating System (ROS) 是一款机器人控制开发系统. 它包含了三维重建、动态仿真、实时场景仿真、实物控制等多种功能,能够最大化地提供机器人开发的各种需求^[6].

ROS 是建立在优班图 (以下简称 Ubuntu) 系统下的一款内置操作系统,但与系统的进程管理没有关系. 它提供各种功能包,然后通过点对点机制建立功能包之间的通讯,搭建控制平台,实现机器人的仿真和实物控制,由此衍生出“节点”,“主题”,“服务”,“消息”等基本概念,如图 1 所示,通过它们之间的配合为控制系统提供数据.

1.1 节点

节点是 ROS 里面的各种可执行文件,各节点之间通过主题,服务来通信. 以此来提高系统的运行效率和可维护性. 节点在系统中具有唯一性,各节点名称不可重复. 这样,使得节点间的通讯亦具有唯一性.

1.2 主题

主题是各节点间传输数据的总线. 通过主题发送的发布者和接收的订阅者之间不需要知道彼此之间的存在. 同一个主题可以拥有多个订阅者,一个主题也可以拥有多个订阅者或者多个发布者,但需要不同的节点发布同样的主题,否则会出现错误.

1.3 服务

服务是用于直接与节点进行通信并且获取反馈. 服务由一对消息组成,一个发送请求消息,另一个接收并应答. 服务是用户自行开发的,存储在 srv 文件夹中,通过编译转换为源码.

1.4 消息

消息用于一个节点向事先定义好的主题发布消息,

从而将数据发送到另一个节点. ROS 里面的所有数据皆可定义为消息. 消息的发送格式具有固定的类型和结构: 功能包名称/.msg 文件名称.

通过节点、主题、服务、消息就可以搭建构建所有进程的控制框架. 如图 1 所示.

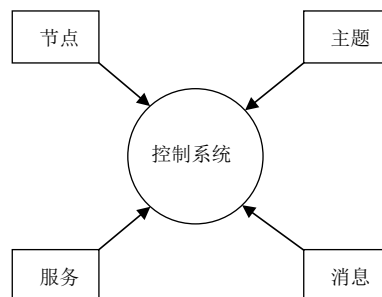


图 1 ROS 的控制框架

2 控制系统算法简述

由于协作机器人在控制过程中需要对各个关节进行精确控制,达到流畅运动的目的,故而在运动的过程中需要一定的控制算法. 本研究的主体算法为机器人运动过程中的路径规划算法. 由于设计方案为输入初始位置和末端位置,进而控制机器人运动. 根据此特点,本控制系统采用基于快速扩展树算法^[7] (Rapidly exploring Random Tree, 以下简称 RRT 算法). RRT 算法是建立在随机树的基础上,通过从根节点随机生长分支节点的方式,进行运动空间中的采样点碰撞检测,不需要进行三维立体建模,解决了三维空间和多条件约束的路径规划问题. 由于其效率高、准确度高、入门容易,因此适用于多自由度机器人在空间中的运动. RRT 算法是以初始位置为根节点,通过随机采样,生成子节点,进而生成一个扩大的随机扩展树,当随机扩展树接触到目标节点时,就可以在随机扩展树里找到一条从根节点到目标节点的路径,即机器人运动的路径规划. 其基本算法如下所示:

Function RRTPlan: BOOL(*env*:environment, *T*:RRTTree, *S_{goal}*: node)

1. Var S_{target} , $S_{nearest}$, S_{new} , *node*
2. While(search time/space remaining) do
3. S_{target} =ChooseTarget(S_{goal})
4. $S_{nearest}$ =Nearest(T , S_{target})
5. IF (Distance ($S_{nearest}$, S_{goal}) < Distance Threshold) then
6. Return true
7. S_{new} =Extend ($S_{nearest}$, S_{target})
8. IF (S_{new} ≠ NULL) then

```
9. T.AddNode( $S_{new}$ )
10. Return false
Function ChooseTarget( $S_{goal}$ : node): ndoe
11. Var  $p$ : real
12.  $p = \text{Random}(0, 1.0)$ 
13. If  $0 < p < aim$  then
14. Return goal
15. Else if  $aim < p < 1.0$  then
16. Return RandomNode ();
```

在随机数的生长过程中 (line 1~10), 初试节点为根节点 S_{init} . 首先, ChooseTarget 函数在工作空间中随机生成点 S_{target} ; 其次, Nearest 函数在随机函数中选择一个距离 S_{target} 最近的中继节点 $S_{nearest}$; 最后, 通过 Extend 函数从 $S_{nearest}$ 向 S_{target} 扩展一段距离 α (α 为 RRT 算法的最小步长), 最终得到新节点 S_{new} . 如果 S_{new} 与障碍物发生碰撞, 则 Extend 函数返回 0, 视为生长失败; 否则将 S_{new} 加入到随机数中. 重复以上步骤, 直到随机数生长到目标点或者距离目标点的距离小于一个阈值, 算法完成.

为加快路径规划的速度, 现将随机树改进为: 先计算出初试位置和末端位置在空间中的中点, 机器人先运动至中点, 在运动至末端位置, 这样可以通过随机概率来决定下一个运动点是否是正确的目标点还是随机点. 事先设定参数 aim , 然后生成 0~1 的随机数 bp , 当 $0 < bp < aim$ 时, 随机数朝目标点运动; 当 $aim < bp < 1$ 时, 随机数的生长方向随机^[8].

通过 RRT 算法, 输入机器人的初试位置和目标位置, 可计算出机器人在运动中各个关节的转动速度、弧度等参数, 并输出.

3 通信接口设计

通信接口模块是搭建上位机电脑与下位机协作机器人的连接. 用于实现两者间控制信息的发布和反馈信息的传送. 由于传统的 ROS 通讯使用 ID 连接的通讯方式, 适用范围狭隘, 仅对于优傲等少数先进的机器人适用, 然而基于控制器局域网^[9] (Controller Area Network, 以下简称 CAN 通讯) 的通讯方式在当下具有普遍适用性, 所以本研究仅调用了 ROS 的路径规划算法与仿真环境, 将其与 CAN 通讯相结合, 建立了一种新型的通讯方式.

将上位机计算出的协作机器人各电机参数, 如: 电机 ID、转动角度、速度等, 调用 CAN 通讯, 将各参数

发送给协作机器人的各个电机上的控制芯片, 进而控制协作机器人. 本通讯方法跳过了 ROS 通讯, 直接建立了与电机之间的通讯, 该方法在原有的稳定性好, 即时性好等特点基础上, 增加了通讯简单、适用性广、移植性高的特点, 可适用于市面上大多数协作机器人.

4 协作机器人总体控制流程

本控制过程分为三个部分: 路径规划运算, 通讯连接, 电机控制. 对于协作机器人的控制, 实质上是对于各个关节处电机转动角度、速度、开关时间的控制. 故而本控制系统控制步骤如下:

(1) 搭建机器人的三维图, 调用 ROS 的环境配置, 设定末端执行器的初始位置.

(2) 在 ROS 的 3D 可视化工具 (Rviz) 界面打开配置好的仿真机器人, 在编辑界面里拖动链条或输入末端执行器坐标, 即可调用 RRT 算法计算出每个关节运动到目标位置所需要的转动角度、速度、加速度、位置等参数, 并且依次将每部分的计算结果发送出去.

(3) 调用 CAN 通讯的方式建立上下位机的通讯, 将计算出的每个关节的转动角度及其他必要参数通过通信接口发送给下位机.

(4) 下位机芯片接收数据, 转换为电机可识别的参数发送给电机, 进而控制机器人运动.

控制流程图如图 2 所示.

5 实验结果与分析

为了验证协作机器人控制系统的实际控制效果, 用 UR5 协作机器人进行了实验验证. 搭建了如下的实验平台.

5.1 仿真实验

在仿真实验中, 搭建控制界面, 在界面里可设置末端执行器的坐标, 通过改变坐标的数值, 进而控制 rviz 中的仿真机器人. 通过仿真界面观察到水平机器人初始位置, 通过在界面拖动或输入末端执行器的坐标轴, 即可改变目标位置, 首先, 使得末端执行器坐标归位, 回到原点. 其次, 在界面改变末端执行器坐标, 使得机器人运动到目标位置, 观察到能够改变机器人各个方向的目标位置并作出相应的路径规划. 如图 3 所示.

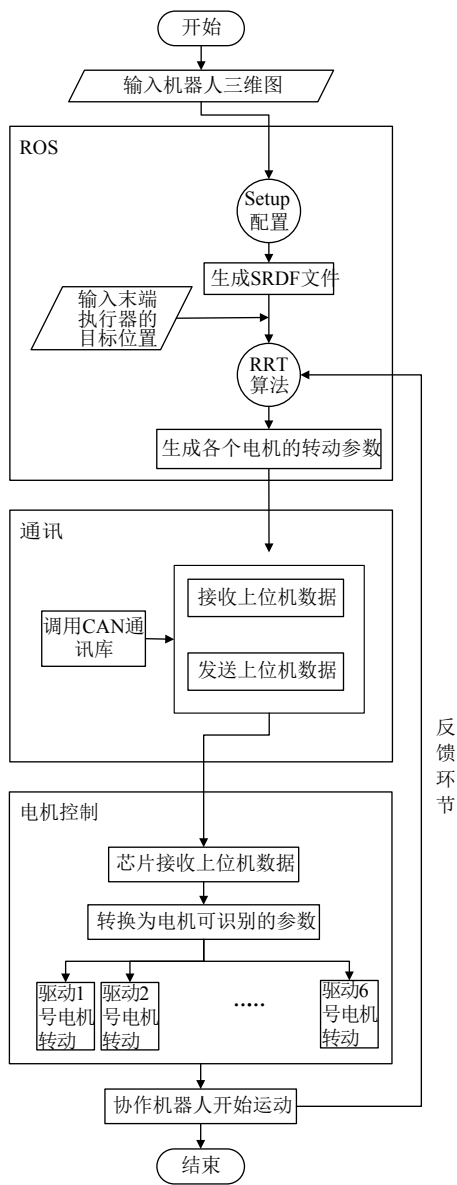


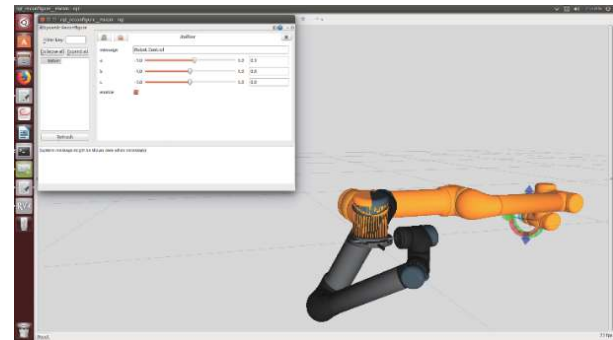
图2 协作机器人控制过程流程图

通过仿真实验结果可知,人为输入末端执行器的目标坐标,在仿真程序中调用RRT算法,可计算出各个关节转动角度,并在仿真界面里体现出来,从而实现对于仿真协作机器人的控制.实验输入的具体坐标,如表1所示.

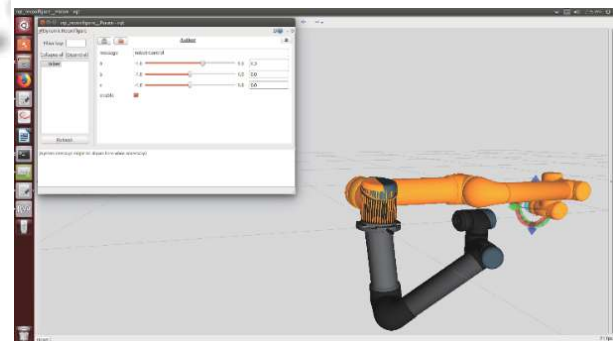
5.2 协作机器人控制实验

实物机器人采用UR协作机器人^[10].上位机界面拖动链条,发送信息,通过通讯发送给机器人内置芯片,芯片通过分析接收的数据,转换为电机可识别的参数,进而控制电机的转速、转动时间等.机器人实验过程如图4所示.上位机发送指令后,下位机接收

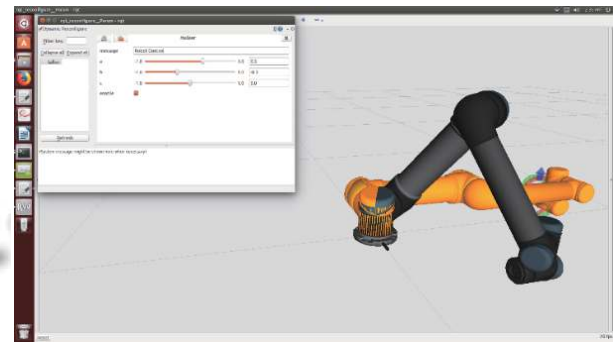
并作出相应的动作,完成路径规划,机器人实体做出相应的动作.



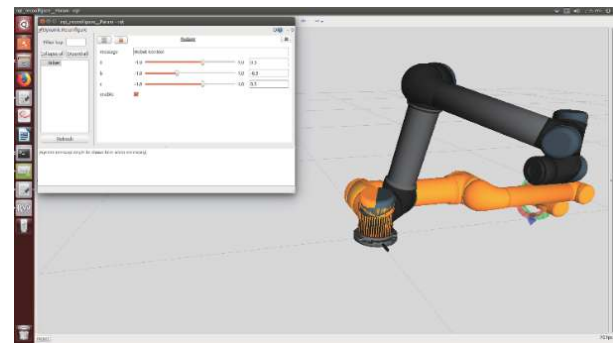
(a) 原点位置



(b) 位置点1



(c) 位置点2



(d) 位置点3

图3 仿真机器人控制实验

表1 输入的末端执行器的坐标

位置点	末端执行器运动到指定坐标
原点	(0, 0, 0)
1	(0.3, 0, 0)
2	(0.3, -0.3, 0)
3	(0.3, -0.3, 0.3)

由于机器人硬件精度误差和 ROS 算法的偏差, 因此在控制系统内对发送的运动参数进行校准. 驱动协作机器人匀速运动, 通过读取各关节绝对位置, 发现机器人并不能运动到指定位置, 误差范围在 2 mm 以内. 经修改控制参数并且完善程序等调试, 可通过上位机界面精确控制机器人的空间运动, 使得机器人有较高的鲁棒性和实时性.



(a) 实物机器人控制实验1



(b) 实物机器人控制实验2

图4 实物机器人控制实验

通过调用 ROS 自带的打印功能, 可在输出界面输出各个关节的速度、时间、加速度等参数. 表2 记录了机器人在输入表1 的末端执行器坐标后运动到指定位置所需时间. 通过输出的时间表明运动较为平缓, 有较好的鲁棒性和实时性.

表2 运动到指定位置点所需时间 (s)

位置点	运动到指定位置所需时间
原点	0
1	1.94
2	1.89
3	2.02

6 结束语

本文基于 Ubuntu 系统下结合 ROS 实现了对于协作机器人的控制. 并且优化了路径规划算法, 改进了 ROS 中上下位机的通讯方式, 建立了在保证原有实时性和鲁棒性前提下适用性广、操作简单的通讯方式, 并搭建了仿真实验和实物控制实验. 由实验可知, 控制系统能够实现协作机器人基本的工作. 有一定的应用价值. 但其他功能的开发, 如: 视觉系统开发、末端抓手开发, 有待进一步的研究, 以达到更佳的使用效果和应用前景.

参考文献

- 1 盛新. 分析和研究影响多轴机器人加工精度的关键因素. 轻工科技, 2016, (12): 71.
- 2 王玮. 应用 Microsoft Visual Studio 2010 开发项目的优势. 现代阅读, 2011, (23): 15-16.
- 3 刘金琨. 滑模变结构控制 MATLAB 仿真. 2 版. 北京: 清华大学出版社, 2012. 15-16.
- 4 尹航, 言勇华. 基于 MRDS 仿人机器人仿真平台设计. 上海交通大学学报, 2010, 44(11): 1529.
- 5 Martinez A, Fernández E. 刘品杰, 译. ROS 机器人程序设计. 北京: 机械工业出版社, 2014. 16-280.
- 6 Hart S, Dinh P, Hambuchen K. The Affordance template ROS package for robot task programming. Proceedings of the IEEE International Conference on Robotics and Automation. Seattle, WA, USA. 2015. 6227-6234.
- 7 Martin SR, Wright SE, Sheppard JW. Offline and online evolutionary bi-directional RRT algorithms for efficient re-planning in dynamic environments. Proceedings of IEEE International Conference on Automation Science and Engineering. Scottsdale, AZ, USA. 2007. 1131-1136.
- 8 尹高扬, 周绍磊, 吴青坡. 基于改进 RRT 算法的无人机航迹规划. 电子学报, 2017, 45(7): 1764-1769. [doi: 10.3969/j.issn.0372-2112.2017.07.029]
- 9 夏继强, 满庆丰, 段丛斌, 等. 计算机与 CAN 通信的一种方法. 测控技术, 2000, 19(7): 58-59. [doi: 10.3969/j.issn.1000-8829.2000.07.021]
- 10 杨霞. 优傲激发人机协作无限可能. 现代制造, 2015, (40): 15.