DOI:10.19344/j. cnki. issn1671-5276.2023.01.044

基于 ROS 和 EtherCAT 的机械臂控制系统

金翰扬,陈富林
(南京航空航天大学 机电学院,江苏 南京 210016)

摘　要:为方便机械臂的使用和后续开发,提出一种基于 ROS 和 EtherCAT 的机械臂控制系统。在上位机 ROS 中构建避障检测、路径规划和通信节点等模块,将机械臂的关节信息用自定义的通信协议串口发送给 STM32;在 STM32 中移植开源的 EtherCAT 协议 SOEM,将机械臂的信息通过工业以太网发送给伺服控制器,完成机械臂的控制,并通过实验验证了该控制系统的可行性。

关键词:机械臂;ROS;EtherCAT;控制系统
中图分类号:TP241　　**文献标志码**:B　　**文章编号**:1671-5276(2023)01-0177-04

Manipulator Control System Based on ROS and EtherCAT

JIN Hanyang, CHEN Fulin
(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: To facilitate the use and subsequent development of the manipulator, a manipulator control system based on ROS and EtherCAT is proposed. In the upper computer ROS, modules such as obstacle avoidance detection, path planning and communication nodes are constructed, and the joint information of the manipulator is sent to STM32 via a user-defined communication protocol serial port. The open source EtherCAT protocol SOEM is transplanted into STM32, sending the information of the manipulator to the servo controller through industrial Ethernet to complete the control of the manipulator. And experiment is conducted to verify the feasibility of the proposed manipulator control system.

Keywords: mechanical arm; ROS; EtherCAT; control system

0　引言

随着科学技术的发展,机械臂不断地融入到人们的生产和生活之中,而机械臂的开发与研究也与时俱进。

ROS(robotic operation system)是 Willow Garage 公司在 2010 年开发的一款开源协议的机器人操作系统。ROS 具有点对点的设计、多语言支持、架构精简,集成度高、组件化工具包丰富和免费且开源这五大特点,使得 ROS 具有分布式结构,每个功能模块可以单独设计、编译,在运行时以松散耦合的方式结合,并且支持多种语言,使得其成为众多机器人开发者的主要工具[1]。

EtherCAT 是由德国倍福公司提出的以以太网为基础的现场总线技术,具有低延时、高响应等特点,支持多种设备连接的拓扑结构[2]。EtherCAT 也符合 OSI 七层模型框架,但只需要物理层、数据链路层和应用层。EtherCAT 为了支持更多种类的设备以及更广泛的应用层,建立了以下应用协议:CoE(基于 EtherCAT 的 CAN 应用协议)、SoE(符合 IEC 61800-7-204 标准的伺服驱动行规)、EoE(EtherCAT 实现以太网)、FoE(EtherCAT 实现文件读取),而从站设备无需支持所有的通信协议,只需选择合适的即可[3]。

因此,本文将结合 ROS 和 EtherCAT 开发一种机械臂控制系统[4],该系统包括机械臂末端的路径规划、位置姿态的控制、人机交互界面、机械臂检测等功能。

1　控制系统整体框架

机械臂控制系统如图 1 所示,系统主要分为两部分[5],一部分是在上位机 ROS 中进行人机交互、路径规划等功能,另一部分是偏向于控制伺服电机等底层的硬件。硬件部分本系统选择是 STM32F407ZGT6 芯片,STM32 和伺服电机驱动器利用 Ethercat 工业以太网总线进行通信,通过 Ethercat 总线可以根据生产环境的具体需求,添加或删减机械臂关节电机的数量或者其他 Ethercat 设备,提高了整个系统的开放性和拓展性。

2　上位机的搭建

ROS 具有分布式结构,每个节点都有各自的任务,通过对各节点间的信息交流,从而达到对机械臂统一控制作用。所以上位机主要是对避障检测、路径规划和通信节点的搭建。

第一作者简介:金翰扬(1997—)男,江苏南京人,硕士研究生,研究方向为智能机器人。

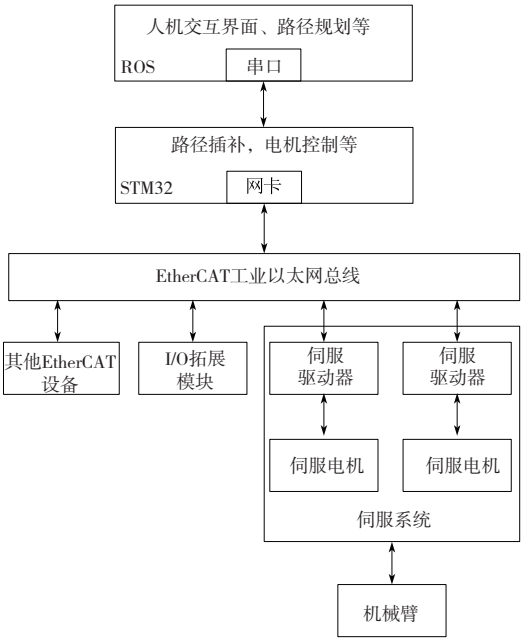


图 1　机械臂控制系统总框架图

2.1　机械臂的避障处理

由于机械臂和障碍物的形状各异,对其进行建模,将大大增加计算量,所以对机械臂的关节采用包围盒技术,用最小的圆柱体将机械臂包裹住[6]。

如图 2 所示是在机械臂 *D-H* 参数模型中的包围盒模型。由于第 1 个连杆固定在地面上,第 2 个连杆绕着第 1 个连杆进行旋转,这两个连杆位置基本不会变化,不用对其进行碰撞检测。第 4 个关节绕其连杆轴线进行旋转,其相对位置不发生改变,第 4 个连杆和第 5 个连杆可以将其看为一个整体。同理,第 6 个连杆和第 7 个连杆也可以看为整体。所以可以将机械臂简化为 3 个圆柱体 C_1、C_2、C_3 相连。

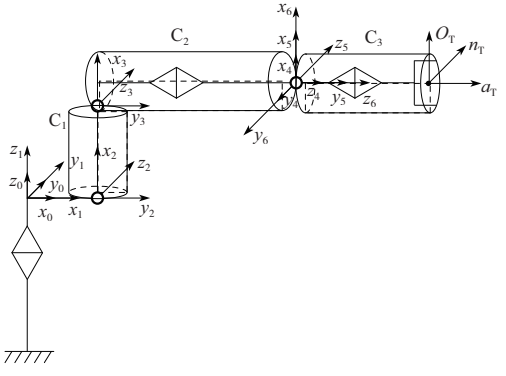


图 2　机械臂包围盒模型

如图 3 所示,用球体将障碍物进行包裹,可以将障碍物的信息简化为球心和半径。但当障碍物为细长状时,仅用一个包围球会将障碍物的体积扩大很多倍,浪费很多空间,不利于路径规划。所以如图 4 所示,采用多个相同包围球进行包围。

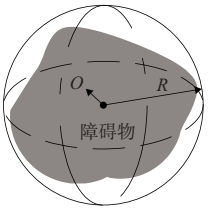


图 3　普通包围球

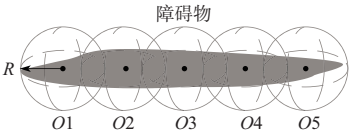


图 4　细长状障碍物包围球

综上,根据各关节的齐次位置矩阵可以计算出各圆柱体的中心线的方程,通过计算中心线到球心的最短距离是否大于半径和来判断是否碰撞[7]。建立 ROS 节点,将此碰撞检测导入 ROS 中。

2.2　路径规划算法

快速扩展随机树算法(rapidly-exploring random trees, RRT)是由 LaValle 教授于 1998 年提出的一种随机采样路径规划算法。RRT 算法采用的是增量式的空间搜索方式,在满足机械臂自身结构限制和环境约束的条件下,解决在静态障碍环境里的路径规划问题[8]。因此在机器人的路径规划方向得到了广泛的应用。

如图 5 所示,在空间 V 中,确定随机树 T_{ree} 起始点 X_{init}、目标点 X_{goal}。在 RRT 算法的计算过程中,在空间 V 下随机采样每个关节角度 $\theta=(\theta_1\theta_2\theta_3\theta_4\theta_5\theta_6)$,通过机械臂运动学正解得到采样点 X_{rand},遍历随机树 T_{ree} 中所有子节点,计算与采样点 X_{rand} 之间的距离,选取距离最小的子节点 X_{near},即

$$X_{near}=\min\|X_{rand}-X\|, X\epsilon T_{ree} \tag{1}$$

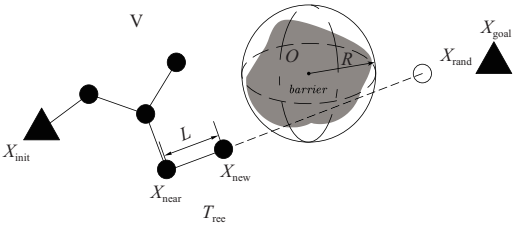


图 5　标准单支 RRT 算法的扩展图

连接 X_{near} 和 X_{rand} 两个子节点,并在线段上截取步长为 L 的点,并对该点进行碰撞和约束检测,若条件满足,将新的节点 X_{new} 加入到扩展随机树 T_{ree} 中,若不符合,则重复采样的过程。新节点 X_{new} 的扩展公式为

$$X_{new}=X_{near}+L\frac{X_{rand}-X_{near}}{\|X_{rand}-X_{near}\|} \tag{2}$$

在重复执行采样 X_{rand} 和扩展随机树 T_{ree} 的过程中,会产生 X_{new} 和 X_{goal} 之间的距离小于所设定的阈值,连接两点,从而完成路径规划。建立 ROS 节点,将算法导入 ROS 中。

2.3　建立 ROS 与机械臂控制器通信

ROS 提供了一种用于 ROS 设备和非 ROS 设备通信方式 rosserial,rosserial 为非 ROS 设备的应用程序提供了 ROS 节点,使其能够通过串口与 ROS 进行数据交互[9]。

由于 ROS 和机械臂控制器采用的是串口通信,串口通信是按字节进行传输的,为了确保数据传输的准确性,将路径点信息进行打包,建立下面的通信协议:

```
union Data
{
    uint8_t buff[32];
    struct Joint_position
{
    uint32_t Header;
    uint32_t receivenumpoints;
    float receivearm1;
    float receivearm2;
    float receivearm3;
    float receivearm4;
    float receivearm5;
    float receivearm6;
}joint_position;
}data;
```

当 STM32 接收到 Header 标志时,表示开始接收新一轮的信息,在这数据中包含此次规划路径的路径点数和 6 个关节的角度信息。

3　机械臂控制器的搭建

3.1　SOEMEtherCAT 主站协议

SOEM(simple open ether CATMaster)是一个开源的 EtherCAT 主站协议库。如图 6 所示,SOEM 是基于 C 语言编写,由若干的模块组成,并采用了分层设计。底层提供了一个由操作系统抽象层(OSAL)和硬件抽象层(OSHW)组成的抽象层。抽象层可以将操作系统和硬件分开,使得 SOEM 理论上可以移植到任意的操作系统和硬件平台上。所以移植的主要目标就是重写 OSAL 和 OSHW,在移植的操作系统和硬件平台上进行具体的实现。由于 SOEM 可以不需要操作系统,所以本文采用直接移植在 STM32F4 硬件平台上舍弃了操作系统层[10]。

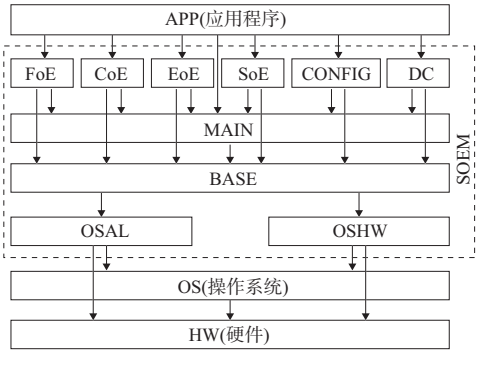


图 6　SOEM 框架图

3.2　定时器的移植

系统抽象层 OSAL 主要包含了时间和定时器还有线程相关的函数接口。由于 EtherCAT 的特性,主站必须有一个系统时钟,EtherCAT 可以根据主站的系统时钟对从站的本地时钟进行插补,从而使得从站的本地时钟近乎统一,达到时钟同步的效果。与此同时 SOEM 采用的是轮询的方式对数据帧进行接收和发送,从站会依次接收到主站发送的数据并且发送返回帧。为防止等待返回帧时间过长,采用了超时机制,当系统时钟超过了设定的时间,就不再接收返回帧从而继续进行下一步的通信。

STM32F4 包含有多个微秒级定时器,将 TIM5 作为系统时钟定时器,设置 1 ms 产生定时器中断。SOEM 的系统时钟采用一个关于 usec、sec 的结构体,当系统时钟定时器产生一次中断时,$usec^{+}=1\ 000$,如果超过 10^6,则 $sec^{+}=1$,$usec^{-}=10^6$,从而完成系统时钟的设定。

3.3　网络驱动移植

硬件抽象层(OSHW)是 SOEM 移植到 STM32 上重点部分,为 STM32 提供了网络服务,是移植工作的核心部分。OSHW 主要是由 oshw 和 nicdrv 两个模块组成。

OSHW 中 oshw 模块的主要功能是实现大小端的转换。STM32 存储方式是小端模式,而网络字节序是大端模式。所以必须在 oshw 中对数据进行大小端的转换,

OSHW 中的 nicdrv 模块主要是实现 EtherCAT 帧的接收和发送。如图 7 所示,在 nicdrv 中设立了 16 个数据缓冲区,当有数据帧需要发送时,在数据缓冲区中先申请一个空间。在申请的数据缓冲区中和要发送的数据帧组成报文通过 MAC 层发送出去;在报文返回时,通过解读报文中的信息,从而确定发送的数据帧所在的数据缓冲区,确保发送和接收的数据帧属于同一个报文。

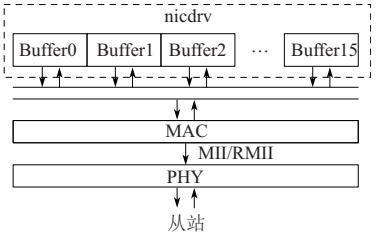


图 7　nicdrv 逻辑结构图

从图 7 可知 nicdrv 模块通过 MAC 层对数据帧进行发送和接收,所以对 nicdrv 模块的移植主要是分 3 个部分:网口的初始化;MAC 层数据帧的发送;MAC 层数据帧的接收。

1)网口初始化:STM32F4 芯片自带以太网模块,包括 DMA 控制器、MII、RMII 还自带外部 PHY 通信的 SMI 接口。STM32F4 要完成以太网通信,也必须外接 PHY 芯片,而本文采用 LAN872A 作为 PHY 芯片通过 RMII 接口与 STM32F4 内部 MAC 相连,通过 DMA 控制器对数据进行接收。所以网口是对包括 MAC、RMII 接口、DMA 控制器、LAN8720A 芯片等的初始化。

2)MAC 层数据帧的发送:数据帧的发送是通过 DMA

控制器将数据帧从内存中搬运到 MAC 中的。为了提高传输效率,DMA 有描述符模式的传输结构,在基于描述符的 DMA 操作中,可以设置多个描述符数据缓冲区,而描述符是对不同的 DMA 配置参数信息,将描述符封装在要传输的数据帧之中,在当前的操作序列完成后,自动设置并启动另一次 DMA 传输。基于描述符的方式为管理系统中的 DMA 传输提供了最大的灵活性。所以在 DMA 传输数据的过程中,CPU 可以同时把数据帧写入到描述符的数据缓冲区中,这样 CPU 与 DMA 协调工作,最大程度地提高了数据帧的发送效率。

3)MAC 层数据帧的接收:STM32 采用轮询的方式接收数据帧,EtherCAT 通信的主站发送数据依次经过从站最后再回到主站。这过程时间短、延迟小,如果采用中断的方式会增加数据的接收时间,影响性能。在轮询的接收到数据后,与 MAC 层数据帧的发送相反,通过 DMA 将从 MAC 传输到描述符的数据缓冲区中,而 CPU 对数据进行处理。

4　平台测试

上位机通过串口连接机械臂控制器 STM32,STM32 通过网线连接伺服驱动器,从而完成整个控制系统的连接。当连接建立后,按下 STM32 上的回零按钮,将此时机械臂的初始状态和 ROS 机械臂可视化模型相匹配,从而建立起联系。打开人机交互界面,连接 ROS,可以进行运动学正解、逆解、点动等操作,如图 8 所示。

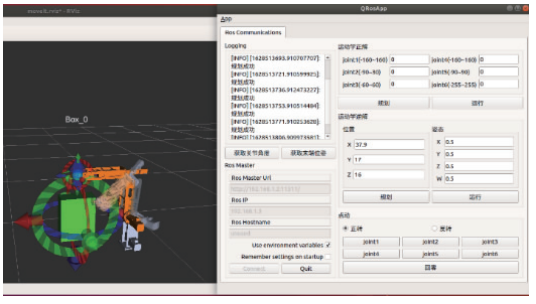
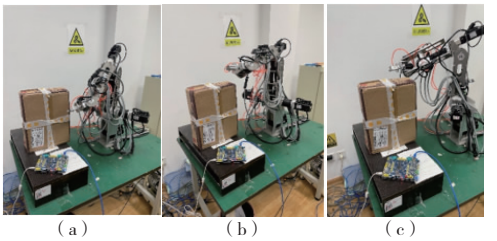


图 8　可视化界面和人机交互界面

如图 9 所示,为了验证机械臂的避障能力,将障碍物信息导入到 ROS 之中,先将机械臂末端的起始点设为障碍物的右侧,将目标点设为障碍物的左侧,结果表明:机械臂能平稳地越过障碍物到达目标点,验证了整个机械臂控制系统的可行性。



(a)　(b)　(c)

图 9　机械臂避障

5　结语

由于 ROS 开发机器人的便利性和 EtherCAT 的可靠性等优点,本文提出了基于 ROS 和 EtherCAT 的机械臂操作系统。在 ROS 中建立避障检测、路径规划和通信节点等模块,通过人机交互界面将机械臂的信息通过串口发送给控制器 STM32,在控制器中建立 EtherCAT 通信从而控制机械臂的运动。整个机械臂控制系统可以根据具体的需求在 ROS 中集成新的模块,也可以方便地添加和减少机械臂关节数,为机械臂后续的开发提供便利。

参考文献:

[1] 李天慧. 基于 ROS 的打磨机器人控制系统研究[D]. 哈尔滨:哈尔滨工业大学,2019.

[2] WANG S, YANG X D, VAN DER GEER J. Development of EtherCAT real-time control system for robot based on Simulink Real-Time[J]. Journal of Computational Methods in Sciences and Engineering,2021,21(1):49-57.

[3] 常玉冬,王超. EtherCAT 主站协议 SoE 的研究与实现[J]. 组合机床与自动化加工技术,2016(11):5-8.

[4] 郝继锋,叶宏,吕广喆. 机器人操作系统 EtherCAT 技术研究[J]. 航空计算技术,2020,50(3):77-81.

[5] 陈前里,刘成良,贡亮,等. 基于 ROS 的机械臂控制系统设计[J]. 机电一体化,2016,22(2):38-40,61.

[6] 马宇豪. 六自由度机械臂避障轨迹规划及控制算法研究[D]. 西安:中国科学院大学(中国科学院西安光学精密机械研究所),2019.

[7] 崔玉霞,段奕林. 基于联合仿真的机械臂最优路径筛选[J]. 机械制造与自动化,2020,49(5):161-164.

[8] 李季,史晨发,邵磊,等. 基于改进 RRT 算法的 6-DOF 机器人路径规划[J]. 计算机应用与软件,2020,37(9):221-226.

[9] 廖炜豪. 基于 ROS 的工业机械臂运动控制器的设计与实现[D]. 广州:华南理工大学,2019.

[10] 王惠娇. 嵌入式平台 EtherCAT 主站的实现[J]. 计算机应用,2018,38(增刊 1):165-169.

收稿日期:2021-09-22