

引文格式: 韩振宇, 郭震宇, 严一钊, 等. 基于机器人-机床制造单元的开放式边缘平台数控系统开发[J]. 航空制造技术, 2025, 68(4): 22-35.

HAN Zhenyu, GUO Zhenyu, YAN Yizhao, et al. Development of open edge platform numerical control system based on robot-machine tool manufacturing unit[J]. Aeronautical Manufacturing Technology, 2025, 68(4): 22-35.

基于机器人-机床制造单元的开放式边缘平台 数控系统开发*

韩振宇¹, 郭震宇¹, 严一钊², 宋剑鑫¹, 金鸿宇¹

(1. 哈尔滨工业大学, 哈尔滨 150006;

2. 中国工程物理研究院材料研究所, 绵阳 621900)

[摘要] 制造业逐步向智能化和柔性化的方向发展。机器人与机床协同工作可以使整个生产过程更加高效灵活,但传统的控制系统及云计算服务无法满足其数据交互的需求。本文面向机器人-机床制造单元,提出了一种开放式边缘平台的服务架构。提出了以实时 Linux 为基础,以 ROS 和 LinuxCNC 开源平台为核心,采用 EtherCAT 主站通信的机器人-机床协同控制系统方案。在控制器与服务器中部署 Fast DDS,实现了不同数据的网络接入,并利用容器技术构建边缘微服务环境,实现了数据的高效交互和传输。最后,搭建了试验平台,完成了系统性能测试。试验验证了所提出方案的可行性。

关键词: 边缘计算; 开放式数控系统; 边缘数据传输; Fast DDS; 工业机器人

DOI: 10.16080/j.issn1671-833x.2025.04.022



韩振宇

教授, 博士生导师, 研究方向为开放式数控系统, 智能数控系统, 重型机床精度模型、故障监测与预测, 纤维缠绕、自动铺丝 CAD/CAM、三维编织技术, 骨骼支架 3D 打印成型。

智能制造作为工业 4.0 时代的关键驱动力,其核心在于实现生产过程的智能化,涵盖自动化设备的互联互通、生产数据的实时监控与分析,以及基于数据驱动的决策支持^[1]。智能制造系统集成化是现代工业发展的重要趋势,它涉及将制造过程中的各个部分通过信息技术进行高度整合,实现设计、生产、管理、服务等各个环节的信息共享和协同工作。传统的云计算存在着数据传输不稳定、传输过程中数据可能丢失、数据处理实时性差、存在网络安全隐患等问题^[2],无法满足工业现场智能化的需求;现有的工业控制系统又相对封闭,数据采集传输困难^[3],用户无法根据需求进行个性化开发^[4]。

边缘计算是一种靠近设备端的计算,通过在数据源附近部署计算能

力,就近实现网络、存储、处理、分析等智能化功能。与云计算不同,边缘计算虽然计算能力较弱,但物理传输距离的减小降低了数据传输延迟^[5],使得数据的本地处理和快速决策成为可能。杨宏青等^[6]提出一种边缘环境下数字化车间制造数据实时融合和协议转换方法,基于边缘计算的思想,构建了制造车间数据实时采集体系架构。华中科技大学的陈家耀^[7]基于现场总线对智能数控系统边缘计算模块协同机制进行了研究,将部分数据处理任务迁移到靠近数据源的边缘设备中进行,利用边缘计算模块提供的额外数据处理能力,扩展数控系统的计算能力。Gupta 等^[8]利用 Trustzone 技术实现了信息技术和操作技术的融合,促进安全的垂直和水平集成。搭建了可信边缘平台,

* 基金项目: 国家自然科学基金(51805116)。

用于诊断管理、固件管理、控制级别管理和威胁配置文件管理的远程生产线维护。Albert 等^[9]介绍了一种名为 Nebula 的轻量级架构,并研究了有效利用边缘资源进行计算和数据存储的方法。浙江大学的冯毅雄等^[10]针对工业控制系统对柔性化、扁平化的需求,融合 5G 网络和边缘计算技术,提出了一种基于 5G 多接入边缘计算的云化 PLC 系统架构,实现了工业现场、企业数据中心、5G 网络和边缘网络的集成。南京航空航天大学尚关卿^[11],提出了一种针对数控机床的边缘端服务平台架构,使用 OPC UA 统一接口技术和 MQTT 协议,实现了边缘端与云端的双向通信。北京邮电大学的李尤^[12]设计了一个边云协同的边缘服务平台,通过组建边缘局域网,获取底层设备的数据,并采用规则处理保证了数据传输的实时性和稳定性。

开放式数控系统强调控制系统的可移植性、可拓展性和可交互性。当前影响较广泛开放式数控架构有美国的 OMAC、欧洲的 OSACA 以及日本的 OSEC 项目,其中 OMAC 的潜在应用范围更广^[13-15]。Xu 等^[16]提出了一种基于 Linux 的嵌入式数控系统的设计,并对其实现的关键技术进行了详细的描述,提高了系统的性能以及可靠性,降低系统开发成本。Hatem 等^[17]基于开放式体系结构控制(Open architecture control, OAC)的可拓展性,结合蚁群算法优化了刀具行程轨迹。华侨大学的张浩晨等^[18]设计了一种基于 PC 机的开放式数控系统,通过实时以太网连接机床硬件实现控制。哈尔滨工业大学的段嘉奇等^[19]基于 LinuxCNC 开发了五轴连续纤维 3D 打印机及其数控系统,并完成了其功能验证。哈尔滨工业大学的富宏亚等^[20]针对机床加工过程中的颤振问题,设计了基于开放式数控系统的颤振识别和在线抑制系统。工业机器人的控制系统大多

为封闭系统或有限开放系统,难以满足研究人员对软件功能开发的需求,为解决这一困难,以斯坦福大学的实验室为主导开发了机器人操作系统(Robot operation system, ROS),ROS 集成了机器人开发所需的多项功能插件,能很好地适应机器人的开发需求^[21]。清华大学的 Ye 等^[22]提出了一种 ROS2 实时性能的优化和评估方法,基于 ROS2 的 DDS 传输机制,利用 Preempt RT 实时内核提高了 ROS2 数据传输的及时性和可靠性。Nwankwo 等^[23]将 ROS 应用于移动机器人中,并实现了多种控制技术,使机器人的操作更直观。哈尔滨工业大学的王雨佳^[24]针对协同生产的场景,搭建了基于 EtherCAT 的集中控制系统,初步实现了基于 ROS 的机器人与基于 LinuxCNC 的机床数控的集中控制。华中科技大学的刘怀兰^[25]提出了一种基于 ROS 和 Unity 的机器人数字孪生系统,并搭建了试验平台,实现了机器人实体与模型的同步运行。

目前针对工业边缘计算上的研究主要集中在较为上层的数据处理和算法模型上,直接面向工业现场的实际生产底层设备的边缘服务平台研究较少,并且大部分研究基于现有的商用控制系统进行数据整合。机器人-机床的协作生产单元通过边缘计算技术,能够实现更高效的生产流程和更灵活的生产调整。结合开放式数控系统的边缘服务器可以实时监测处理生产数据,优化生产计划,减少生产中断,并实现快速的工艺调整。机械臂除了可以进行上下料以外,还可以实现打磨、焊接等柔性生产。通过集成的更加先进的传感器,在边缘平台上实时监测产品质量,减少废品率。综上所述,开放式数控系统和边缘服务器在机床-机器人生产单元的应用,有着诸多优点,是智能制造发展的重要方向。

本文针对智能车间中的智能化

柔性化生产需求,提出一种工业机器人与数控机床组成的生产协作制造单元的开放式数控系统和边缘服务平台架构,并搭建了试验平台。上述方案重点解决了生产制造单元内部的信息交互和运动控制的问题,并通过试验验证了其可行性。

(1) 面向机器人-机床制造单元,设计开放式数控系统及其边缘服务的整体网络架构;基于所设计的控制系统方案,使用开源控制平台搭建控制系统;基于开源主站实现控制系统与底层硬件的通信,实现机器人与机床的运动控制。

(2) 基于开源通信中间件构建边缘接入网络,实现机器人、数控机床与边缘云服务器的数据交互;搭建边缘节点内部服务网络,并基于该网络构建边缘服务平台,实现基于边缘服务器计算资源优势的功能部署。

(3) 针对机床与机器人组成的协作生产制造单元,设计试验方案并搭建试验平台,测试边缘数据网络的实时性及可靠性,在此基础上验证开放式数控系统和边缘服务平台的可行性。

1 开放式数控系统的开发

智能工厂等概念的提出推动着工业生产现场设备逐渐走向智能化。由于成本和技术的制约,难以实现在本地进行数据存储和生产优化等功能。因此,越来越多的工厂开始部署边缘服务器,发挥边缘计算的优势,来提升工业现场计算能力。这对于现场制造单元提出了新的要求:首先是底层设备控制系统的开放性,采用封闭控制系统的传统生产设备之间相互独立,功能固定,无法根据用户的需求重新修改和扩展系统的功能,难以实现统一规划和协调生产,因此需要重新搭建基于边缘服务平台的开放式数控系统来取代设备原有的专用控制系统;开放式数控系统采用模块化结构设计,允许用

户对数控系统进行定制化的二次开发,快速进行系统功能扩展,实现生产过程中的机床协作生产与集中管控,降低系统升级和维护的成本,提高生产效率。

1.1 开放式数控系统及边缘服务平台基本架构和控制系统基础环境选择

开放式数控系统及边缘服务平台的核心是不同设备和服务间的数据传输,因此需要选择合适的数据传输技术来构建边缘通信网络,实现不同数据和平台之间的数据传输。由于目前工业机器人和数控机床是工业现场最主要的生产设备,本文面向工业机器人和数控机床组成的协作制造生产单元,设计开放式数控系统及边缘服务平台基本架构,如图1所示。

其中设备层由底层设备及其控制器组成。通过工业现场以太网由边缘设备层控制器直接控制,以保证控制实时性;设备层控制器基于实时操作平台,搭载开放式数控系统和开源主站,与硬件设备层直接通信以实现底层设备的运动控制和数据采

集。边缘层由边缘服务器搭载边缘服务平台,采集设备层控制器产生的实时数据,对数据流进行初步处理并实现边缘数据存储;通过对设备层控制器发布任务,实现边缘集中控制,同时实现状态监测。边缘服务器通过云服务软件接口和互联网将可共享大数据上传至云服务器,利用云服务更强的计算能力和更大的存储空间,实现所需的大数据模型建立和数据存储。基于该架构,本文对设备层和边缘层部分功能进行具体开发研究。

本文采用 X86 架构工控机的全软件架构进行开发,用 PC 机作为工控机。由于过高的延迟会导致控制系统错误甚至有可能引起机械故障,因此控制系统对于响应延迟具有较高的要求,需要以实时操作系统 (Real-time operating system, RTOS) 作为底层开发环境,及时对触发信号进行处理。Linux 系统相对于目前常用的 Windows 系统有更好的实时性,并且在软件开发上也有很好的通用性,因此本文将采用 Linux 系统

进行控制系统的开发。Linux 系统常见的发行版,例如 Ubuntu、Debian 和 LMDE 等都搭载非实时内核,在响应任务时会产生较大延迟,不能满足控制系统对于实时性的需求。因此需要对原内核进行改造,将实时线程的周期性响应延迟降低到 10 μ s 以内^[26]。理论上单内核实时改造方案 RT-Patch 是向 Linux 添加实时性的最精简、最干净的方法^[27],有着支持的硬件平台较多,易于维护等优点。因此引入 Preempt-RT 补丁,利用 Linux 内核的可抢占特性,在不违背开源版权协议的条件下,通过改造内核实时调度策略,直接修改内核源代码,实现实时化。

EtherCAT 协议在同步精度、稳定性和传输速度等方面有较好性能,同时又具有灵活性高和硬件需求低的优点,只需要有标准的以太网接口就可以实现主从站之间的通信,因此适用于本文使用的普通 PC 机。更重要的是 EtherCAT 拥有可用的开源主站方案。综上所述,本文将选用 EtherCAT 技术协议作为机器人与机床控制系统的底层伺服通信协议。为保持整体系统的开放性,同时降低开发成本,选用社区开源的 EtherCAT 主站方案。

1.2 工业机器人控制系统的开发

在目前已有的开源机器人开发平台中,ROS 是应用最广泛的框架之一,适用于工业机器人、移动机器人和仿生机器人等各类机器人的控制系统开发。ROS 是一种松散耦合的软件框架,采用分布式通信机制实现各功能模块进程之间的通信。ROS 的各个功能模块基于对各类开源和共享项目的支持与融合,这些项目的源代码由不同的机构共享和发布,比如 Rviz、Moveit! 等,并支持 C、C++、Python 等多种编程语言。本文基于 ROS 平台开发的六轴工业机器人控制系统架构如图2所示。

如图3所示,通过 ROS 集成的

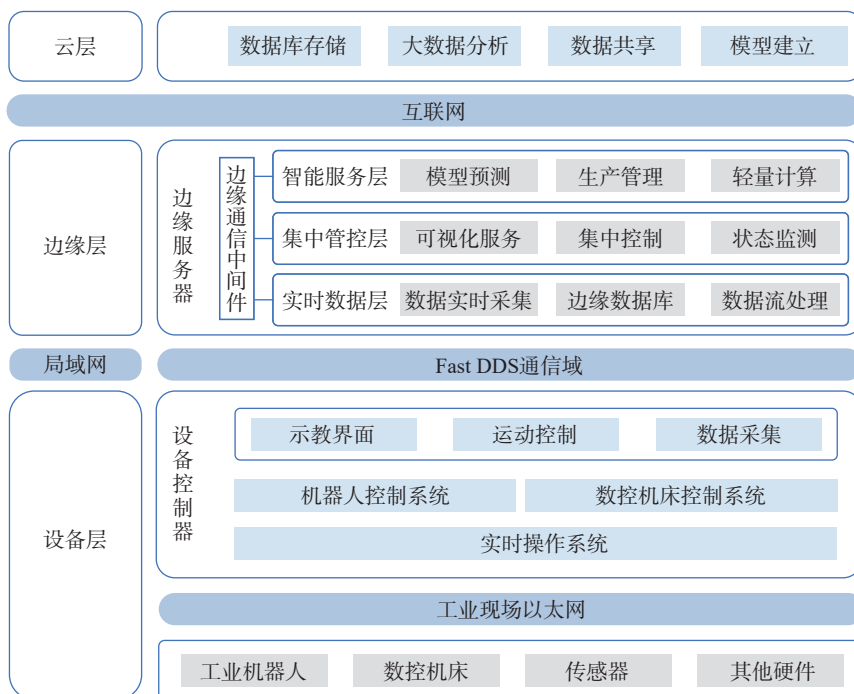


图1 开放式数控系统及边缘服务平台基本架构

Fig.1 Basic architecture of open control system and edge service platform

Rviz 可视化插件,可以显示机器人当前状态,反馈当前运动情况。如图 3 (a) 所示,可在右侧操作界面上通过设置目标位姿,进行 Rviz 的可视化运动仿真;另一方面,通过 Rviz 的功能包模型输出与仿真,也验证了机器人模型参数模块的正确性。机器人

控制系统的核心在于机器人的运动规划,ROS 平台集成的 Moveit! 运动规划插件(图 3 (b)),可实现机器人正逆运动学求解并在 Rviz 上发布机器人控制指令。通过 Moveit! 运动规划插件,还可以进行运动规划处理,在三维空间判别碰撞,确定各轴运动

参数。本文使用开放式运动规划库 (Open motion planning library, OMPL) 进行机器人运动轨迹规划。

本文选择 SOEM (Simple open EtherCAT master) 作为开源 EtherCAT 主站方案以实现机器人的控制。与其他主站方案相比,SOEM 主站开发难度较低^[28],且 ROS 已经实现了对 SOEM 的功能包支持。SOEM 主站基于 EtherCAT 协议,其配置核心是创建 IOMap 映射地址,设置从站过程数据对象 (Process data object, PDO) 信息,提供上层数据映射地址。完成 PDO 地址配置后,设置机器人的 EtherCAT 同步周期为 4 ms,以第一个从站 J0 关节轴伺服器的时钟作为参考基准设置,配置 SOEM 主站的 DC 分布式时钟同步,以保证主从站的同步精度。为防止由于主站延迟过大导致机器人发生碰撞,需在使用前测试主站性能。本文采用 Tcpcdump 抓包工具,获取基于 EtherCAT 协议的报文信息^[29]。启动主站连接机器人后,以第一次报文发送时间刻度为基准,抓包结果如表 1 所示。设定 SOEM 主站报文 4 ms 循环一次,即插补周期为 4 ms,实际循环周期的最大延迟为 7 μs,处于 50 μs 以内。此外,Preempt-RT 补丁下定时器的延迟和抖动也是系统实时性能所面临的主要挑战。运行 cyclictst 命令测得 Preempt-RT 补丁

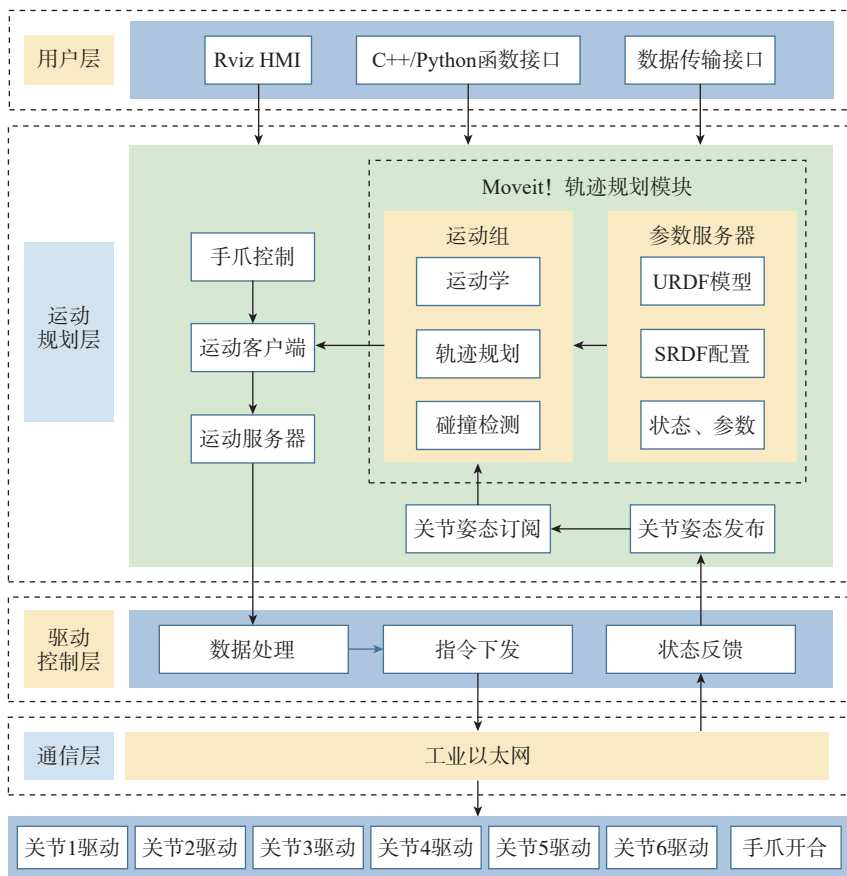
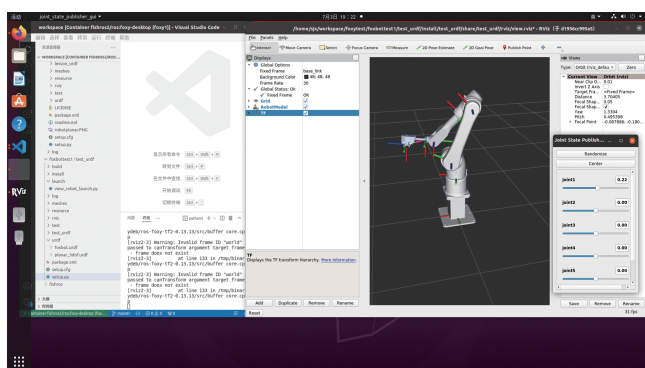
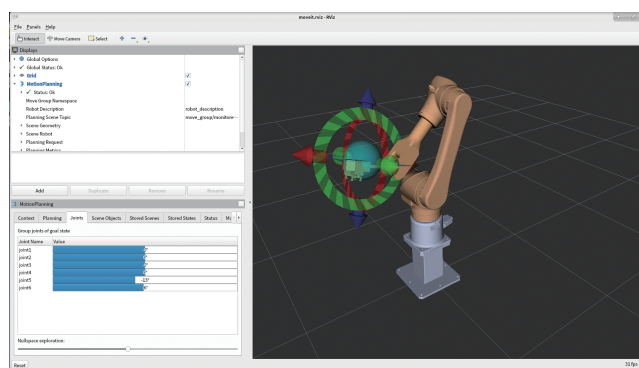


图 2 工业机器人控制系统架构图

Fig.2 Architecture diagram of industrial robot control system



(a) 可视化运动仿真



(b) 运动规划模块

图 3 Rviz 界面
Fig.3 Rviz interface

下一段时间内定时器的延迟数据如表 2 所示,结果表明主站实时性较好,可用于机器人控制。

在完成 SOEM 主站与从站通信后,还需将主站与运动规划模块相关联,以实现机器人各轴的运动控制。构建 ROS 控制机器人运动的基本代码逻辑如图 4 所示。

在大多数情况下,机器人的自动复位很可能导致机器人与其他设备产生碰撞,引起设备损坏。为解决机器人复位问题,在控制指令映射时,加入了等待使能环节。在控制系统使能前,机器人保持原有状态,在控制系统使能后,再将目标指令下发至底层控制目标。

1.3 数控机床控制系统开发

LinuxCNC 是一款开源的数控软件,最高支持 9 轴联动控制和实时轨迹规划,可以操作真正的伺服系统,且具有高度的通用性和可拓展性,被广泛用于机械加工领域。因此,本文基于 LinuxCNC 开放式数控平台,开发 DYNA 三轴数控铣床的控制系统。数控系统内置了直线圆弧插补算法模块,可满足三轴机床的运动需求。在数控加工过程中,产生大量的加工过程数据有存储等复杂需求,需要有效的传输手段。IgH 主站对 EtherCAT 的实现程度更为完整,能够支持 CoE、SoE 等多种通信协议。同时集成 IgH EtherCAT 主站与 LinuxCNC 纯软件数控系统的机床重复定位精度可达 $5 \mu\text{m}$ ^[30],延迟在 $50 \mu\text{s}$ 以内^[31],可以很好地满足工业机器人与数控机床等的加工精度和实时响应要求。因此选用 IgH 主站来控制数控机床。数控机床通过 IgH 主站将各轴运动参数发送给驱动器,同时发送信号给 IO 模块,完成对数控机床的控制。同时基于数采模块,将传感器信号从 EtherCAT 接入,实现加工过程数据采集。基于 LinuxCNC 的控制系统主要架构如图 5 所示。

表 1 SOEM 主站报文时间数据

Table 1 SOEM indicates the packet time of the primary station

ms

报文发送时间	报文接收时间	循环时间/ $\times 10^{-4}$	报文周期	延迟
0.000	0.176	1.76	—	—
3.993	4.141	1.48	3.993	-0.007
7.985	8.130	1.45	4.002	0.002
11.989	12.136	1.47	4.004	0.004
15.983	16.136	1.53	3.994	-0.006

表 2 Preempt-RT 补丁下定时器的延迟

Table 2 Delay of timer under Preempt-RT patch

μs

时间间隔	最小延迟	平均延迟	最大延迟
4000	0	1	18
8000	0	1	32
12000	1	1	19
16000	0	1	20

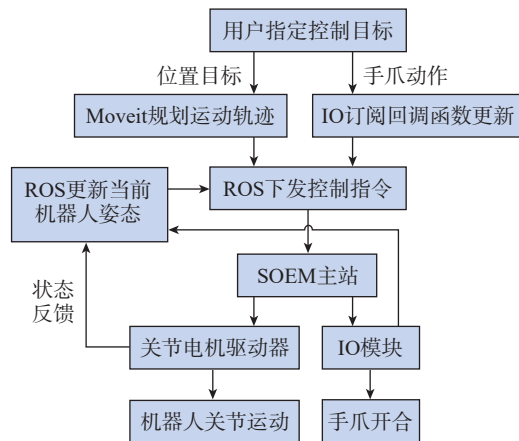


图 4 ROS 控制机器人运动代码逻辑

Fig.4 Logic of ROS code for controlling robot movement

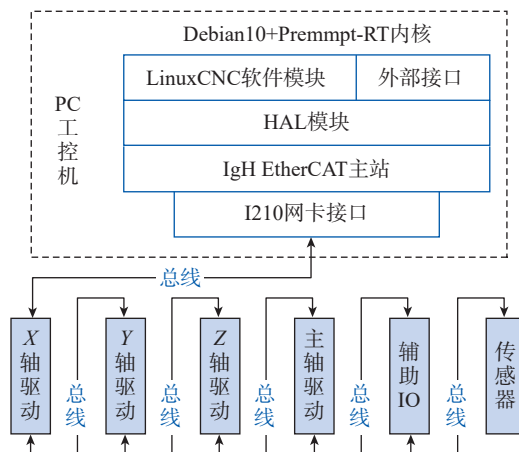


图 5 基于 LinuxCNC 的控制系统主要架构

Fig.5 Main architecture of control system based on LinuxCNC

与 SOEM 主站类似, IgH 主站的从站配置模块也需要基于 EtherCAT 协议, 设置从站 PDO 信息。但是 LinuxCNC 无法直接与 IgH 主站建立连接, 需要将 IgH 主站功能封装为 LinuxCNC 的 HAL 组件进行开发。HAL 模块是 LinuxCNC 为了进一步扩大自身的重新配置特性, 作为分层放置在 EMCOT、EMCIO 等模块和硬件之间的驱动程序。建立 IgH 主站的配置文件, 被放置在外部的 HAL 模块里, 通过配置文件进行编译链接, 并通过引脚连接建立 LinuxCNC 的主站通信机制。数控机床具有多个 IO 控制量和数采模块, 因此需和驱动器从站一同配置 PDO 信息。在 EtherCAT 中, 主站通过检测状态机跃迁来实现主从站的规范运行^[32], 跃迁过程如图 6 所示。主站检测到的从站状态共有 4 种, 分别是初始化状态 (Init)、准备运行状态 (Pre-Op)、安全运行状态 (Safe-Op) 和完全运行状态 (Op)。在完成该状态设置且从站检查无误后, 由主站请求进入下一状态。在数控机床的主从站进行周期性数据交互的过程中, 通过配置 IgH 主站的 DC 分布式时钟同步, 来保证主从站同步精度, 避免因主从站异步响应产生的数控机床运行抖动和异响问题。

完成主从站代码配置后, 启动主站连接从站, 使用抓包工具获取基于 EtherCAT 协议的报文信息, 以第一次报文发送时间刻度为基准, 抓包结果如表 3 所示。为使机床加工精度更高, 选择较短的插补周期, IgH 主站报文 1 ms 循环一次, 实际循环周期的最大延迟为 6 μ s, 处于 50 μ s 以内, 满足性能需求。

在完成 IgH 主站的 HAL 模块开发并编译至 LinuxCNC 动态链接库中后, 编写加载文件和配置文件, 建立 LinuxCNC 与该 HAL 模块的连接, 实现 LinuxCNC 对数控机床的实际控制。

2 边缘数据网络搭建及服务平台开发

数据的传输是构建大型分布式系统的关键。针对边缘集群架构的分布式多服务器通信, 以及目前通用的协议尚不完善的现状, 中间件可以提供具有高可行性的解决方案。中间件可以将不同操作系统提供应用的接口进行标准化, 协议统一化^[33], 使计算机硬件、操作系统、应用服务和网络有机结合在一起, 保证了消息中间件在工业集群中应用的可操作性。本文选择 Fast DDS 作为边缘服务平台的数据传输中间件。DDS 技术能够高效地处理微型设备的管理任务, 同时支持大型、高性能的传感器网络和对时间敏感的密集控制回路。此外, DDS 还具备与云端数据

交换的能力, 在国防、汽车制造和仿真技术等对数据传输速率有高要求的领域中, DDS 已经得到了广泛的应用。Fast DDS 是基于 DDS 协议的通信中间件, 有较好的性能和代码开放性, 同时在设备间通信上也可以完全代替现在常用的 OPC UA 协议。Fast DDS 具有较低的延迟和较高的吞吐量^[34], 可以满足机床和机器人数据传输的要求。

2.1 Fast DDS 通信环境部署

为将机器人 ROS 控制器、数控机床 LinuxCNC 控制器和边缘云服务器, 均接入中间件 Fast DDS 节点, 以便实现边缘端机器人与机床设备的通信, 需先部署可适配 ROS、LinuxCNC 和 Fast DDS 的系统环境。本文采用 Debian10 系统作为 LinuxCNC 设备层控制器系统环境; 采用 Ubuntu20.04 作

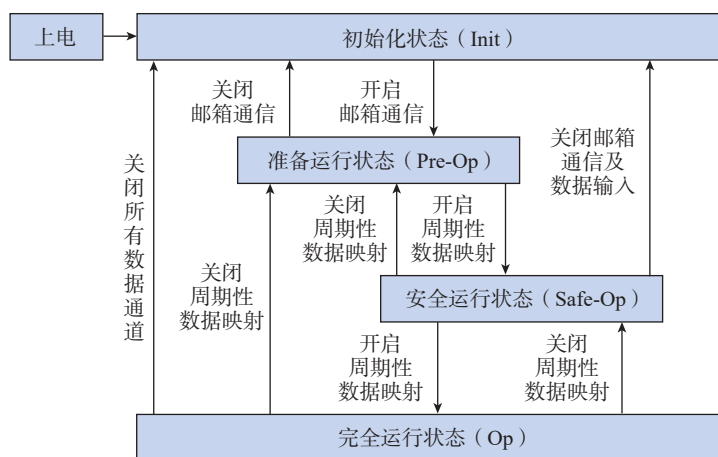


图 6 EtherCAT 状态机跃迁过程

Fig.6 Process of EtherCAT state machine transition

表 3 IgH 主站报文时间数据

Table 3 IgH indicates the packet time of the primary station

ms

报文发送时间	报文接收时间	循环时间/ $\times 10^{-5}$	报文周期	延迟
0.000	0.079	7.9	—	—
1.003	1.084	8.1	1.003	0.003
1.999	2.007	7.8	0.996	-0.004
2.998	3.077	7.9	1.002	0.002
4.000	4.080	8.0	1.006	0.006
5.006	5.085	7.9	0.999	-0.001
6.005	6.080	7.5	0.994	-0.006

为 ROS 设备层控制器和边缘服务器的系统环境,部署过程如图 7 所示。

成功部署 Fast DDS 环境后,在工作区连接 Fast DDS 库,启动 Fast DDS 节点进行通信。将环境部署完成的多台设备接入局域网后, Fast DDS 通信环境部署成功。最后,采用进程间通信的方式将基于 ROS 的机器人控制系统和基于 LinuxCNC 的机床控制系统接入 Fast DDS 节点,实现在边缘网络中的数据交互。

2.2 微服务网络架构设计

微服务架构将应用拆分为多个自治小应用,每个应用可以独立实现特定的业务功能。由于开发环境、语言和运行依赖可能不同,通常使用虚拟机 (hypervisor) 或容器 (container) 技术进行隔离。虚拟机消耗较多资源,而容器技术更轻量,采用共享宿主主机内核的方式可以快速启动,节省内存^[35-36]。Docker 是常用的开源容器技术,支持拉取或构建镜像,创建运行容器微服务,其基本逻辑如图 8 所示。本文基于 Fast DDS 提供的基于 Ubuntu 的镜像程序构建应用。为实现设备层与边缘层数据交互,需设计微服务网络架构。Docker 提供 4 种网络模式:网桥 (bridge)、主机 (host)、无网络 (none)、容器 (container)。网桥为默认模式,自动配置 IP;主机模式直接映射宿主机网络;无网络模式无额外配置;容器模式共享网络。根据 Fast DDS 通信需求,采用主机网络与网桥网络混合模式,满足通信需求。设计基于容器的微服务网络架构,使用 Docker-Compose 进行容器编排管理,实现状态监测与数据采集服务的“一对多”订阅模式。设计基于容器的微服务网络架构如图 9 所示。

2.3 边缘服务功能开发

边缘集中管控服务是针对工业机器人和数控机床的自动化生产任务管控。本文通过结合 Qt5 开发框架和 Fast DDS 通信环境开发人机界

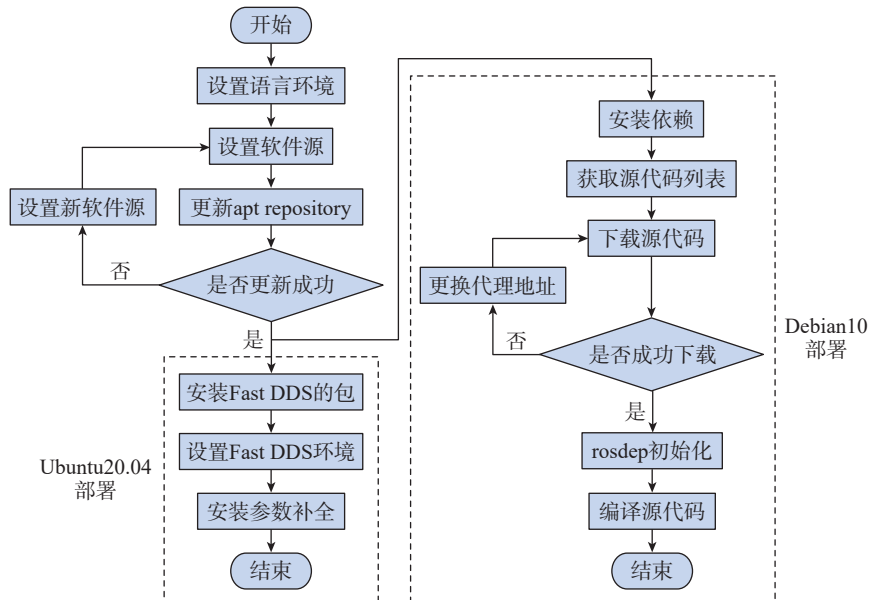


图 7 Fast DDS 环境部署流程

Fig.7 Process for deploying the Fast DDS environment

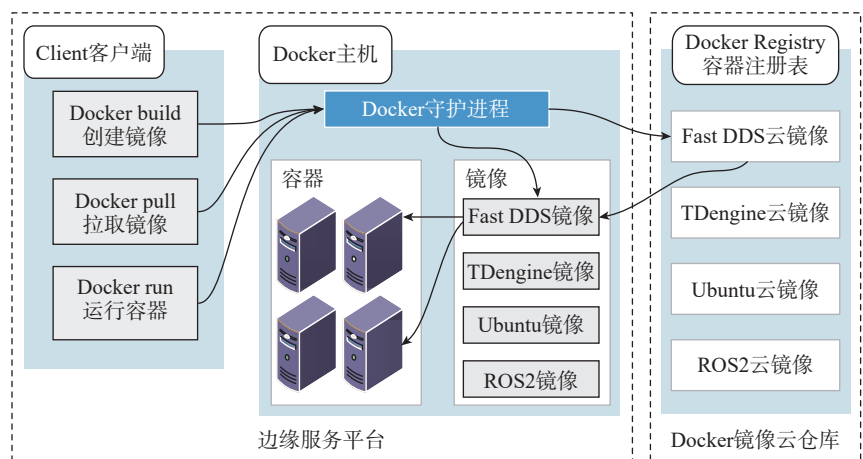


图 8 基于 Docker 创建容器

Fig.8 Creating containers based on Docker

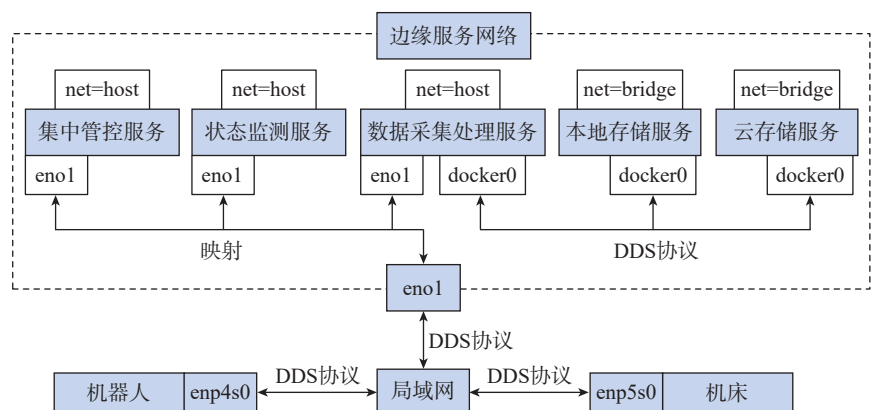


图 9 基于容器的微服务网络架构

Fig.9 Container-based microservices network architecture

面,结合设备层网络的接入,实现制造过程的集中管控、数据传输处理和人机交互。通过 xserver 显示器挂载实现图形界面显示。配置完成后的集中管控服务应用界面包括导航栏、状态显示、任务选项和生产状态显示,提供机器人和机床状态监控及紧急停止功能,支持任务分配和执行选项,搭建的界面如图 10 所示。数据采集及在线监测服务用于实时采集和监测机床加工数据,采用直接映射主机 IP 的网络模式。对于切削力和加速度等实时性要求较高的数据,通过 Fast DDS 创建数据接收和发布节点,实现六个通道的数据采集。由于传感器数据频率高,采用超采样技术,设置 10 kHz 采样频率,对数据进行分割和转换处理,实现实时数据处理和发布。通过在 Fast DDS 镜像容器中部署 PlotJuggler 插件(Qt 的数据可视化工具)实现传感器数据的实时可视化,加工过程实时数据监测界面如图 11 所示。

3 试验与分析

在完成设备层控制系统和边缘数据网络搭建及边缘服务开发后,需要对各部分功能进行试验验证和性能测试,并针对不满足需求的性能进行 QoS 参数优选。通过机器人-机床制造单元协作上下料加工的试验,验证搭建的机器人与机床控制系统与边缘服务的可行性。

3.1 平台搭建及性能测试

试验平台硬件组成主要包括工业机器人及其控制器、数控机床及其控制器、边缘服务器等。试验平台组成如图 12 所示。机床加工过程中,通过数据采集设备获得力和加速度信息。

本文基于 Fast DDS 主要传输 3 类数据,机床加工过程数据、任务指令数据和接收到的机床和机器人运行状态的反馈数据。其中,任务指令数据只在每次任务下发时,进行一次数据传输,其主要强调传输数据



图 10 集中管控服务应用界面

Fig.10 Centralized management service application interface

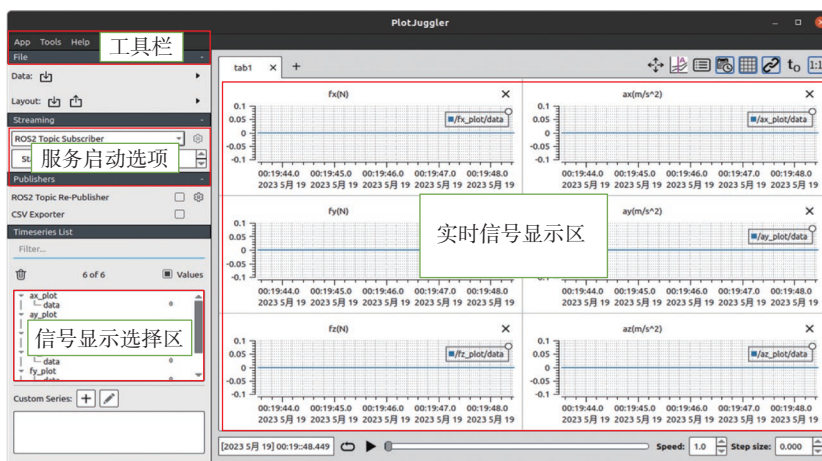


图 11 加工过程实时数据监测界面

Fig.11 Processing process real-time data monitoring interface



图 12 试验平台硬件总体组成

Fig.12 Overall composition of the experiment platform hardware

的可靠性,对延迟要求不高,因此采用 Reliability 配置项目。而机床和机器人运行状态的反馈数据是实时更新的,因此需要进行持续的数据传输。机床与机器人的状态一般是持续性的,不会出现数据的突然跳动,所以数据传输的丢包率只需要满足人为观测需求即可。基于默认配置,开启 20 个状态反馈通道,进行 5 次数据传输测试,主要测试丢包率和延迟,丢包率通过对比发布与订阅的数据量获得,延迟通过对最后一条数据的发布与订阅时间戳作差获得。为保证获得准确的延迟数据,本文利用 NTP (Network time protocol) 协议使局域网内的设备层控制器与边缘服务器的系统时钟达到同步,其同步精度可以达到毫秒级。测试结果如表 4 所示。5 次测试的数据传输丢包率均小于 0.1%,且延迟均在 10 ms 内,满足观测要求。

对于数据采集服务接收的加工过程中的传感器数据,未来可能有数据分析、模型建立等使用需求,其传输需要保证较高的完整性。同时为保证使用者对于一些诸如刀具磨损状态等在线监测数据实时使用,数据传输的延迟也应保持较低水平,一般要求延迟在毫秒级。传感器数据为持续传输数据,本文结合 Depth 和 History 配置项目设置,调整发布者和订阅者缓存数据的队列大小,对不同的队列大小组合进行丢包率和延迟的试验,再从中选取较好的参数进行设置。传感器数据在不同的队列大小组合的传输情况下,其丢包率和延迟试验结果如图 13 所示。可以看出订阅队列大小是影响延迟与丢包率的主要因素,而发布队列对丢包率和延迟的影响较小。由图 13 (b) 可以看出,不同队列大小组合的延迟均在 15 ms 以内,整体具有较好的实时性。当订阅队列大小保持一致时,如果发布队列长度过小,会导致较高的丢包率。当发布队列增大到一定值

后,对减少丢包率的影响有限。当发布队列大小一致时,随着订阅队列的增大,丢包率逐渐减小,但延迟逐渐增大。在订阅队列较小的情况下,丢包率和延迟的变化较为剧烈。当订阅队列大小超过 10 后,丢包率基本保持不变,说明此时队列大小已经足够容纳数据,丢包率不再显著下降。

根据上述试验结果,选取发布队列大小和订阅队列大小均为 10 的组合,此时数据传输中丢包率较低,且延迟约为 7 ms。在边缘智能化服务中,对于一些实时性要求不高的服务,其监测延迟在几秒甚至几十秒内,毫秒级延迟完全满足需求^[37]。而对于一些实时性要求较高的服务,如铣削颤振实时监测,其监测延迟要求在 100 ms 内,平均处理分析时间在 85 ms 左右,7 ms 的延迟同样满足高实时性要求^[38]。在线监测服务接受的传感器数据需进行预处理,然后进行数据时域图的实时绘制。该过程

与数据采集服务相比,会占用较多的处理时间,影响其数据传输性能,导致出现如图 14 所示的数据图像不连续。同时,加工过程中可能出现数据突然抖动的情况,这一般会反映某种加工过程状态,当丢包率过大时,可能导致该状态无法被观测。

为了优化数据传输性能,进行了不同队列大小组合下的试验,以确定 Depth 和 History 配置项目参数。丢包率和延迟试验结果如图 15 所示。试验结果表明,订阅队列大小对丢包率和延迟的影响比发布队列大小更大。随着订阅队列的增大,丢包率减小,而延迟增大。

由于发布队列大小对性能参数的影响不大,选择的与数据采集服务一致的发布队列大小为 10。为满足在线数据监测需求,避免实时数据图出现明显的数据丢失,要求丢包率控制在 2% 以内,延迟保持在毫秒级。初步确定的订阅队列大小在 100~1000 之间,进一步的细分试验得到图 16 的试验结果,确定了更优的订阅队列大小为 500,此时丢包率约为 1%,延迟约为 500 ms,满足了在线监测的需求。参数优选后的在线监测实时输出时域图如图 17 所示。

3.2 制造单元控制系统及边缘服务可行性验证

为验证集中管控服务及在线监测服务可行性,针对机器人-机床制造单元,设计实现机器人与机床的协

表 4 状态数据传输性能测试
Table 4 Test of status data transmission performance

序号	丢包率/%	延迟/ms
1	0.06	6.84
2	0.09	7.01
3	0.05	6.87
4	0.06	6.55
5	0.02	6.21

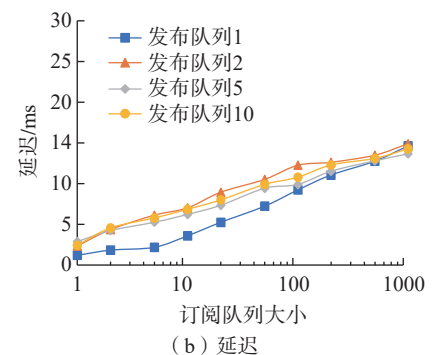
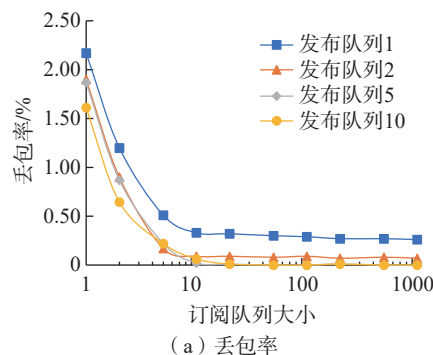


图 13 加工过程数据的丢包率和延迟试验结果

Fig.13 Experimental results of packet loss rate and delay of processing data

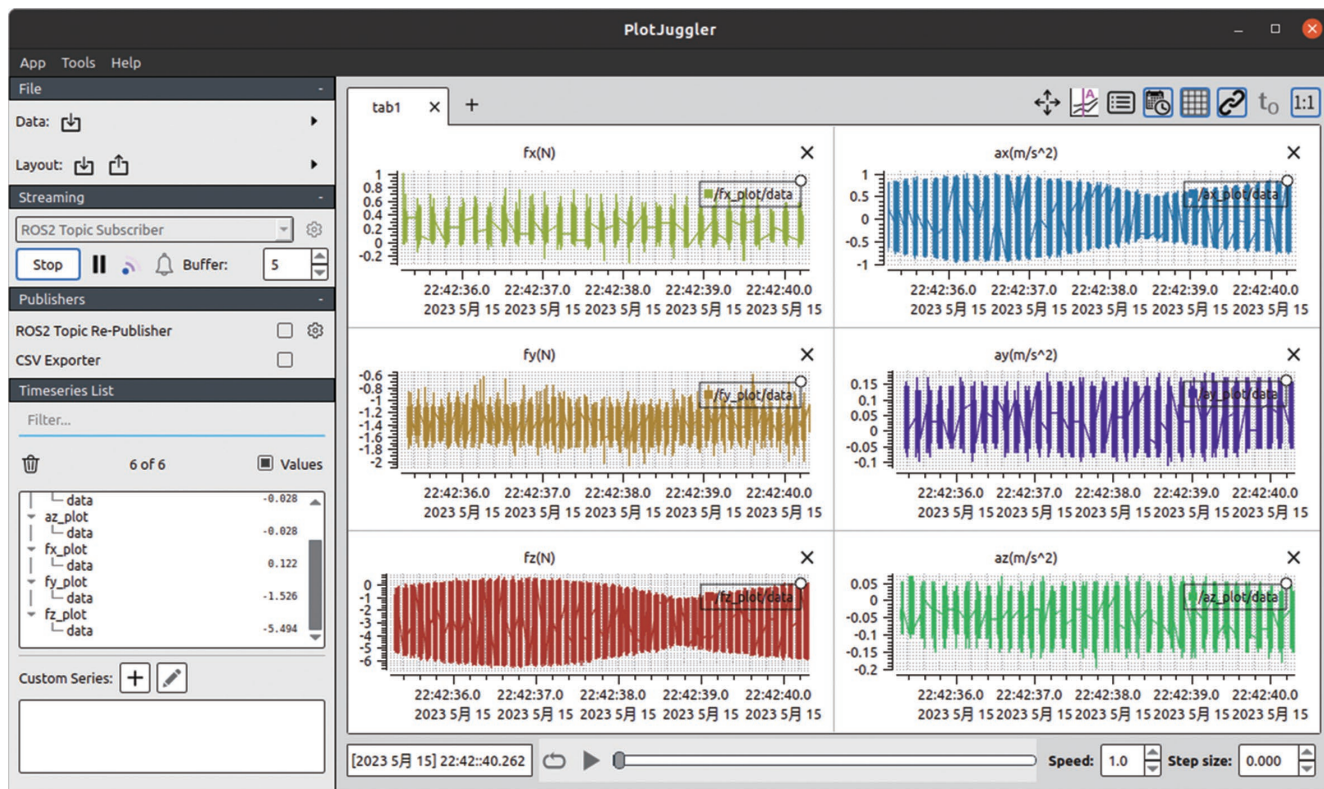
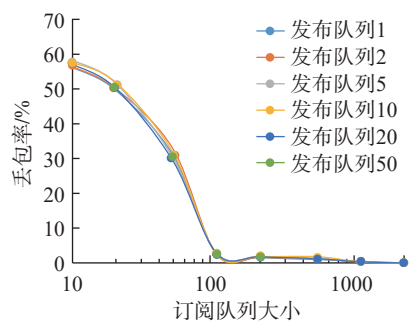
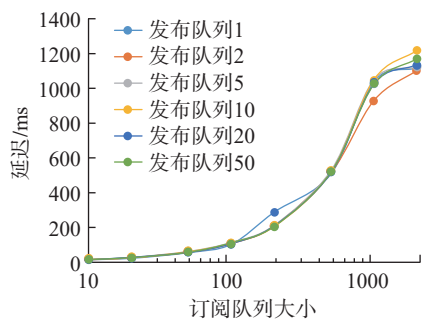


图 14 数据图像不连续的情况

Fig.14 Situation of the discontinuous data



(a) 丢包率



(b) 延迟

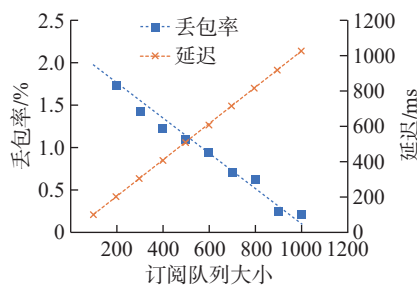
图 15 加工过程数据的丢包率和延迟试验结果
Fig.15 Experimental results of packet loss rate and delay of processing data

图 16 发布队列为 10 的丢包率和延迟

Fig.16 Packet loss rate and latency of the publish queue is 10

作上下料加工过程。由于现场机器人工作环境复杂,为避免直接进行两点之间的规划导致求解时间过长甚至求解失败,采用正运动学与逆运动学结合的控制方式。同时,为避免逆运动学抓取时产生碰撞,在机器人模型中添加了碰撞检测模型。机器人与机床协作制造过程试验情况如图 18 所示。在集中管控服务下发任务后,机器人与机床控制系统顺利控

制设备完成上下料加工生产的任务。加工过程中边缘集中管控服务窗口显示如图 19 所示,可以有效展示机床与机器人当前状态。在线监测服务运行情况如图 20 所示,能够有效观测机床加工时,刀具的切入切出情况,信号数据时域图保持连续,且观测时未感到明显延迟。

4 结论

(1) 本文提出了一种面向机床-机器人制造单元的开放式控制系统和边缘服务平台的整体架构;确定了以 ROS 和 LinuxCNC 开源平台结合 EtherCAT 的控制系统开发方案;结合 ROS 和 SOEM 开源主站完成了机器人控制系统的开发,利用 LinuxCNC 及 IgH 开源主站实现了对机床的控制。

(2) 在设备层控制器与边缘服务器搭建了 Fast DDS 通信环境,实现了机床、机器人的控制系统的接入;基于 Fast DDS 镜像,实现容器内边缘服

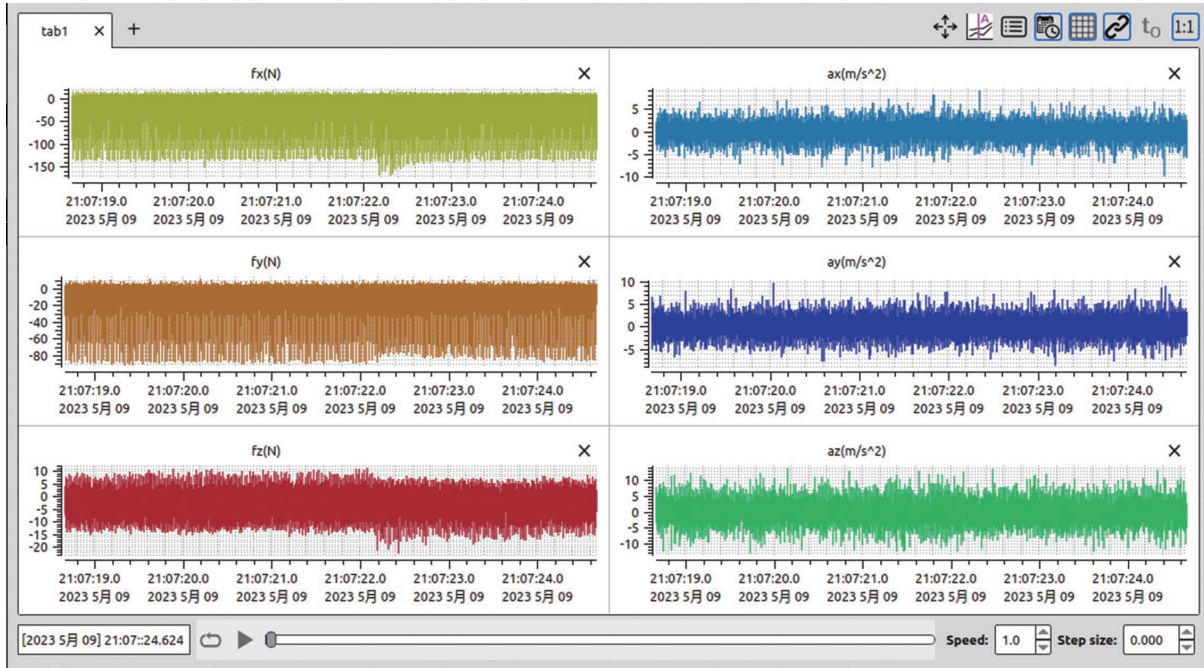
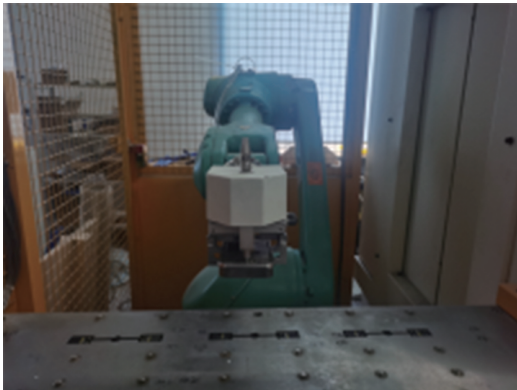


图 17 参数优选后的连续图像
Fig.17 Continuous image after parameter optimization



(a) 机器人取料



(b) 机器人上料



(c) 机床加工



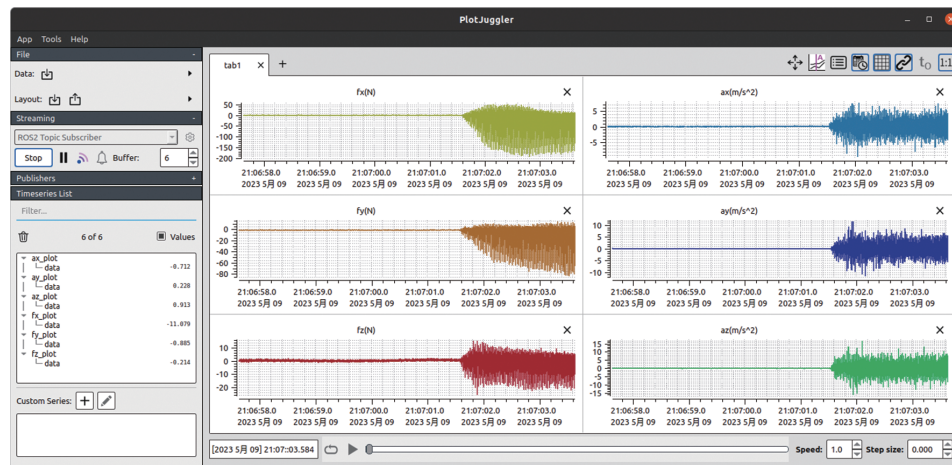
(d) 机器人下料

图 18 机器人-机床协作制造过程
Fig.18 Robotic-machine tool collaborative manufacturing process

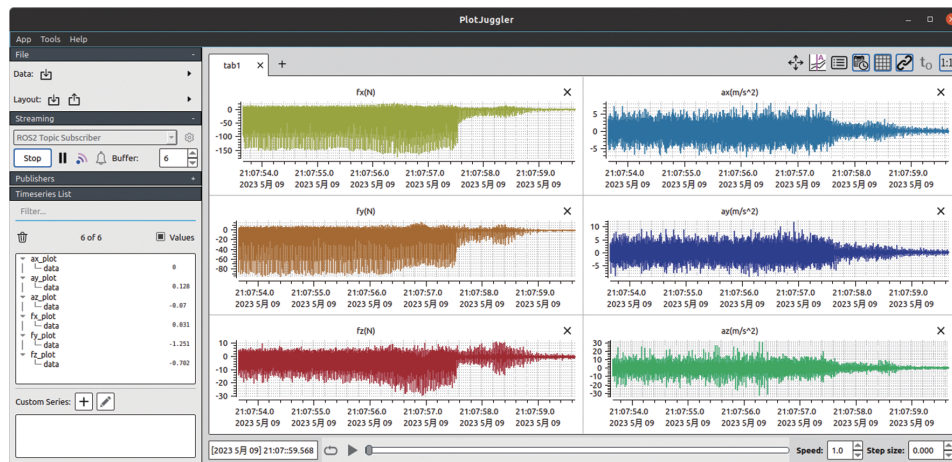


图 19 集中控制服务运行情况

Fig.19 Centralized control of service health



(a) 刀具切入监测



(b) 刀具切出监测

图 20 在线监测服务运行情况

Fig.20 Online monitoring of service operation health

务的网络部署,并与设备层 Fast DDS 节点关联,完成了集中管控服务、数据采集服务、在线监测服务的开发。

(3)搭建了试验平台,进行了数据传输性能测试和参数优选,满足了数据参数需求;设计了机器人与机床协作上下料的制造加工试验,完成了协作制造过程,验证了控制系统及边缘服务的功能可行性。

参考文献

- [1] 周济. 工业强基推进产业基础高级化为制造强国奠定坚实基础[C]. 2022国家制造强国建设专家论坛. 宁波, 2022.
- ZHOU Ji. Strengthening industrial base, promoting the upgrading of industrial base and laying a solid foundation for manufacturing a strong country[C]. 2022 National Manufacturing Power Construction Expert Forum. Ningbo, 2022.
- [2] PRAKASH V, WILLIAMS A, GARG L, et al. Cloud and edge computing-based computer forensics: Challenges and open problems[J]. *Electronics*, 2021, 10(11): 1229.
- [3] 孟博洋, 李茂月, 刘献礼, 等. 机床智能控制系统体系架构及关键技术研究进展[J]. *机械工程学报*, 2021, 57(9): 147-166.
- MENG Boyang, LI Maoyue, LIU Xianli, et al. Research progress on the architecture and key technologies of machine tool intelligent control system[J]. *Journal of Mechanical Engineering*, 2021, 57(9): 147-166.
- [4] 路松峰. NC-Link 标准及应用案例[J]. *世界制造技术与装备市场*, 2021(1): 33-36.
- LU Songfeng. NC-Link standard and its application cases[J]. *World Manufacturing Engineering & Market*, 2021(1): 33-36.
- [5] LAROUI M, NOUR B, MOUNGLAH, et al. Edge and fog computing for IoT: A survey on current research activities & future directions[J]. *Computer Communications*, 2021, 180: 210-231.
- [6] 杨宏青, 周子颖, 魏乐愚, 等. 面向边缘计算的数字化车间制造数据实时融合方法[J]. *南京航空航天大学学报*, 2024, 56(1): 80-87.
- YANG Hongqing, ZHOU Zijie, WEI Leyu, et al. Edge computing oriented real time fusion method of digital workshop manufacturing data[J]. *Journal of Nanjing University of Aeronautics & Astronautics*, 2024, 56(1): 80-87.
- [7] 陈家耀. 智能数控系统边缘计算模块关键技术研究[D]. 武汉: 华中科技大学, 2023.
- CHEN Jiayao. Research on key technologies of edge calculation module in intelligent CNC system[D]. Wuhan: Huazhong University of Science and Technology, 2023.
- [8] GUPTA S. An edge-computing based Industrial Gateway for Industry 4.0 using ARM TrustZone technology[J]. *Journal of Industrial Information Integration*, 2023, 33: 100441.
- [9] ALBERT J, RYDEN M, KWANGSUNG O, et al. Nebula: Distributed edge cloud for data intensive computing[C]. 2014 IEEE International Conference on Cloud Engineering. Boston, 2014.
- [10] 冯毅雄, 杨晨, 胡炳涛, 等. 基于5G多接入边缘计算的云化PLC系统架构设计与应用[J]. *计算机辅助设计与图形学学报*, 2024, 36(1): 33-46.
- FENG Yixiong, YANG Chen, HU Bingtao, et al. Architecture design and application of cloud PLC system based on 5G and multi-access edge computing[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2024, 36(1): 33-46.
- [11] 尚关卿. 面向智能制造车间的数控机床边缘端服务平台技术研究[D]. 南京: 南京航空航天大学, 2020.
- SHANG Guanqing. Research on edge service platform technology of NC machine tool for intelligent manufacturing workshop[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2020.
- [12] 李尤. 基于边缘计算的工业物联网云平台架构的设计与实现[D]. 北京: 北京邮电大学, 2020.
- LI You. Design and implementation of cloud platform architecture for industrial internet of things based on edge computing[D]. Beijing: Beijing University of Posts and Telecommunications, 2020.
- [13] GOLDEN E, HERRIN J, GILLESPIE J W, et al. Open module architecture controllers(OMAC)[J]. *Modern Machine Shop*, 1996, 68(11): 160-162.
- [14] JEAN V, OWEN C, GILLESPIE J W. Opening up controls architecture[J]. *Manufacturing Engineering*, 1995, 115(11): 53-60.
- [15] SAWADAC, AKIRA O. Open controller architecture OSEC-II: Architecture overview and prototype systems[C]. 1997 IEEE 6th International Conference on Emerging Technologies and Factory Automation Proceedings. Los Angeles, 1997.
- [16] XU X M, LI Y, SUN J H, et al. Research and development of Open CNC system based on PC and motion controller[C]. *International Conference on Information, Computing and Telecommunications (ICICT)*. Harbin, 2012.
- [17] HATEM N, YUSOF Y, KADIR A Z A, et al. A novel integrating between tool path optimization using an ACO algorithm and interpreter for open architecture CNC system[J]. *Expert Systems with Applications*, 2021, 178: 114988.
- [18] 张浩晨, 谢明红, 熊金宇, 等. 基于PC及TwinCAT3 PLC的开放式数控系统设计与实现[J]. *制造业自动化*, 2020, 42(3): 75-78, 89.
- ZHANG Haochen, XIE Minghong, XIONG Jinyu, et al. Design and implementation of open CNC system based on PC and TwinCAT3 PLC[J]. *Manufacturing Automation*, 2020, 42(3): 75-78, 89.
- [19] 段嘉奇, 郝明晖, 秦若森, 等. 五轴连续纤维3D打印机及其数控系统开发[J]. *航空制造技术*, 2022, 65(6): 68-75.
- DUAN Jiaqi, HAO Minghui, QIN Ruosen, et al. Development of 5-axis continuous fiber 3D printer and its numerical control system[J]. *Aeronautical Manufacturing Technology*, 2022, 65(6): 68-75.
- [20] 富宏亚, 金鸿宇, 韩振宇. 智能加工技术在切削颤振在线抑制中的应用[J]. *航空制造技术*, 2016, 59(7): 16-21.
- FU Hongya, JIN Hongyu, HAN Zhenyu. Application of intelligent machining technology in on-line suppression of cutting chatter[J]. *Aeronautical Manufacturing Technology*, 2016, 59(7): 16-21.
- [21] ROS Introduction [EB/OL]. [2018-08-08]. <http://wiki.ros.org/ROS/>.
- [22] YE Y L, NIE Z G, LIU X J, et al. ROS2 real-time performance optimization and evaluation[J]. *Chinese Journal of Mechanical Engineering*, 2023, 36(1): 144.
- [23] NWANKWO L, FRITZE C, BARTSCH K, et al. ROMR: A ROS-based open-source mobile robot[J]. *HardwareX*, 2023, 14: e00426.
- [24] 王雨佳. 数控机床与工业机器人集中控制系统研究[D]. 哈尔滨: 哈尔滨工业大学, 2022.
- WANG Yujia. Research on centralized control system of CNC milling machine and industrial robot[D]. Harbin: Harbin Institute of Technology, 2022.
- [25] 刘怀兰, 赵文杰, 李世壮, 等. 数字孪生车间机器人虚实驱动系统构建方法[J]. *中国机械工程*, 2022, 33(21): 2623-2632.
- LIU Huailan, ZHAO Wenjie, LI Shizhuang, et al. Construction method of virtual-real drive systems for robots in digital twin workshops[J]. *China Mechanical Engineering*, 2022, 33(21): 2623-2632.

[26] ARM J, BRADAC Z, KACZMARCZYK V. Real-time capabilities of linux RTAI[C]. 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016. Brno, 2016.

[27] SIRO A, EMDE C. Assessment of the realtime preemption patches (RT-Preempt) and their impact on the general purpose performance of the system[EB/OL]. [2007-04-05]. <https://www.osadl.org/fileadmin/events/rtlws-2007>.

[28] 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.

HAO Jifeng, YE Hong, LV Guangzhe. Research on EtherCAT technology based on robot operating system[J]. Aeronautical Computing Technique, 2020, 50(3): 77-81.

[29] ZHANG J B, MA M, WANG P, et al. Middleware for the internet of things: A survey on requirements, enabling technologies, and solutions[J]. Journal of Systems Architecture, 2021, 117: 102098.

[30] 王斌, 茹浩磊, 王云峰, 等. 基于LinuxCNC数控系统和EtherCAT总线的复杂异型孔激光加工机床的开发与应用[J]. 组合机床与自动化加工技术, 2019(12): 10-14.

WANG Bin, RU Haolei, WANG Yunfeng,

et al. The development and application of laser processing machine for the complex special-shaped hole based on LinuxCNC and EtherCAT[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2019(12): 10-14.

[31] 张克华, 马佳航, 田林晓, 等. 开源IgH EtherCAT主站在LinuxCNC开放式数控系统中的应用研究[J]. 制造技术与机床, 2018(6): 191-195.

ZHANG Kehua, MA Jiahang, TIAN Linxiao, et al. Application research on the open source IgH EtherCAT master in open numerical control system LinuxCNC[J]. Manufacturing Technology & Machine Tool, 2018(6): 191-195.

[32] 马春敏, 康存锋, 黄旭东, 等. 基于Linux的EtherCAT主站的研究[J]. 制造业自动化, 2011, 33(8): 78-82.

MA Chunmin, KANG Cunfeng, HUANG Xudong, et al. Research on the EtherCAT master under linux[J]. Manufacturing Automation, 2011, 33(8): 78-82.

[33] 谢锴, 颜钢锋. 基于EtherCAT的多轴运动控制系统设计[J]. 组合机床与自动化加工技术, 2017(2): 68-72.

XIE Kai, YAN Gangfeng. Design of motion control system based on EtherCAT[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2017(2): 68-72.

[34] Fast DDS vs Cyclone DDS [EB/

OL]. [2021-07-21]. <https://www.eprosima.com/index.php/resources-all/performance/fast-dds-vs-cyclone-dds-performance>.

[35] GOOUTAM K, PRASAD S G. Docker & containers, the future of microservices[J]. Journal of Global Economy, Business and Finance, 2021, 3(4): 23-25.

[36] 杨保华, 戴王剑, 曹亚仑. Docker技术入门与实战[M]. 北京: 机械工业出版社, 2015.

YANG Baohua, DAI Wangjian, CAO Yalun. Docker primer[M]. Beijing: China Machine Press, 2015.

[37] GUO L, YU Y X, QIAN M G, et al. FedRUL: A new federated learning method for edge-cloud collaboration based remaining useful life prediction of machines[J]. IEEE/ASME Transactions on Mechatronics, 2023, 28(1): 350-359.

[38] HAN Z Y, ZHUO Y, YAN Y Z, et al. Chatter detection in milling of thin-walled parts using multi-channel feature fusion and temporal attention-based network[J]. Mechanical Systems and Signal Processing, 2022, 179: 109367.

通信作者: 金鸿宇, 副研究员, 博士生导师, 研究方向为开放式数控系统、切削过程控制、铣削颤振识别与抑制、铣削变形控制、智能加工。

Development of Open Edge Platform Numerical Control System Based on Robot-Machine Tool Manufacturing Unit

HAN Zhenyu¹, GUO Zhenyu¹, YAN Yizhao², SONG Jianxin¹, JIN Hongyu¹

(1. Harbin Institute of Technology, Harbin 150006, China;

2. Institute of Materials, Chinese Academy of Engineering Physics, Mianyang 621900, China)

[ABSTRACT] With the gradual development of the manufacturing industry towards intelligence and flexibility, the collaborative work of robots and machine tools can make the entire production process more efficient and flexible. However, traditional control systems and cloud computing services are unable to meet the demands of their data interaction. This paper proposes an open edge platform service architecture for robot-machine tool manufacturing units. The proposed solution is based on real-time Linux, with ROS and LinuxCNC open-source platforms as the core, and employs EtherCAT master station communication for the robot-machine tool collaborative control system. Fast DDS is deployed in the controller and server to achieve network access for different data, and container technology is utilized to build an edge microservice environment, enabling efficient data interaction and transmission. Finally, an experimental platform was set up, and system performance tests were conducted. The experiments verified the feasibility of the proposed solution.

Keywords: Edge computing; Open numerical control system; Edge data transmission; Fast DDS; Industrial robots

(责编 大漠)