

虚实结合的低延时工业机器人监控调试系统^①

禹鑫焱^② 应皓哲^③ 郑 坤 王 犇 欧林林

(浙江工业大学信息工程学院 杭州 310023)

摘 要 为了解决工业机器人监控系统监控手段单一、监控信息不全面、故障排查困难、远程监控延时高等问题,开发了一种虚实结合的低延时工业机器人监控调试系统。基于远程过程调用(RPC)框架和快速 UDP 互联网连接(QUIC)协议,实现了“云-边-端”协同的分布式监控系统,能够跨语言跨服务器地调用资源并降低系统延时。利用 Beego 框架和 Vue 以前后端分离的方式开发人机交互良好的 Web 监控系统。在后端对多类机器人的通信协议和多种数据库接口封装,实现机器人数据的采集和存储。通过采集的实时数据驱动前端的虚拟模型,以虚实结合的模式对设备运行状态进行监控。在人机安全方面,基于 YOLO 目标检测算法以虚拟围栏的形式为机器人划定三维的工作空间,实时检测人机的空间位置关系,保障人身安全。为保证远程控制的安全性,采用虚拟调试技术预先对机器人的运行动作在虚拟环境中进行仿真,通过仿真验证安全性后生成离线文件下发给机器人执行,保证远程调试的安全。通过实验验证了系统的可行性。

关键词 云-边-端协同; 远程监控; 虚实结合; QUIC 协议; 虚拟围栏

0 引 言

机器人作为一种高效率、智能化的生产工具,应用范围涉及工业制造、航空航天和医疗护理等各个领域,在我国具有很高的使用占比。其中工业机器人与“工业 4.0”提倡的“无人工厂”、“智能制造”等理念完美贴合,是实现实体制造行业从传统的劳动密集、资源密集向数字化、网络化、智能化转变的重要推动力量^[1-2]。但是随着工业机器人使用场景的增多,工作场景内设备量的增加,工作环境愈发复杂。随之带来了安全隐患增加、故障排查困难等问题,大量设备的同时运行也增加了网络的负荷、恶化了网络环境,对搭建有效、稳定且低延时的监控系统提出了新的挑战^[3]。如何提前预防和有效解决工业机器人自身运行安全以及人机安全问题,同时减

少网络延时和波动对系统性能的影响,具有很高的应用和研究价值。

目前国内外很多学者对于工业生产环境下的监控系统进行了研究。文献[4]针对工业机器人提出了一种生命周期风险评估的方案。文献[5]基于云计算对喷涂机器人设备进行远程监控以及诊断系统,能够匹配到不同的故障类型。文献[6]结合机器人运动学和实时监测数据实现多机器人协同的动态视觉监测。文献[7]基于 Unity 3D 虚拟现实引擎开发 B/S 架构下的三维虚拟监控系统,能实时再现工业机器人的真实状态。文献[8]利用具有空间分辨率的触觉地板作为监测空间安全的传感器,通过地板颜色提示空间的安全区域。在监控系统的通信方式上,文献[9]通过 WiFi 模块采集传感器节点数据,并以以太网传输。文献[10]以消息队列遥测传输协议(message queuing telemetry transport, MQTT)

① 国家重点研发计划(2018YFB1308001),浙江省教育厅一般科研(GZ20731030014)和浙江省自然科学基金(LY21F030018)资助项目。

② 男,1979 年生,博士;研究方向:机器人控制与规划;E-mail: yuxinyinet@163.com。

③ 通信作者,E-mail: yinghaozhe716@163.com。

(收稿日期:2021-04-26)

进行客户端和服务端间的通信,传输机器人角度数据。文献[11]利用 TCP/IP 协议的远程通信网关进行远程工业设备控制。文献[12]基于以太网通过 Socket 和 Ethernet KRL 接口,实现机器人与上位控制系统的实时数据交互。

目前研究所设计的远程监控系统大多将重心放在对设备运行状态的监控上,缺乏对于人机安全的监控手段。同时网络波动以及延迟等负面因素的影响,导致监控系统的整体性能大打折扣。因此高效的工业机器人多类安全问题监控方案具有较高的研究价值。

本文在此研究背景下面向工业机器人生产环境,设计了基于“云-边-端”架构的虚实结合的低延时工业机器人监控调试系统,在“云”层以浏览器/服务器模式(browser/server, B/S)结构设计了交互控制界面,实现对工业机器人生产环境的远程监控。在机器人数据监控的基础上,以现场画面实时监控结合虚拟仿真环境同步运行的形式对设备运行状态多维度监控;同时结合部署在“边”层带算力的嵌入式设备,对生产环境下的人和运行的工业机器人进行实时图像处理,实现人机安全监测。根据所评估的风险实时做出警告信息和安全决策。最后构建了实验平台,以一系列实验对虚实结合的低延时工业机器人监控调试系统进行功能验证和性能分析,验证系统的可行性。

1 系统需求及总体框架设计

本文设计的虚实结合的低延时工业机器人监控调试系统主要用于解决工业机器人生产环境中传统监控系统监控手段单一、监控信息不够全面、故障时故障原因排查困难以及远程监控高延时等问题。从系统硬件构成和系统软件结构两方面对系统进行总体方案设计,系统总体框架如图 1 所示。

系统硬件构成方面,所需的系统硬件主要包括部署在生产环境内网的机器人及其控制台、带算力和丰富接口的嵌入式设备、摄像头和工业路由器及高性能的云服务主机。对系统硬件结构的设计包括对设备进行选型、系统环境构建和连接调试等操作。

系统的软件结构方面,为了增加系统的及时性和安全性,采用边-云协同的系统架构进行软件设计。在“云”层,为了给用户提供可视化界面以显示多维度的监控信息,以及提供简单易操控的交互控制界面,采用 Vue.js 对交互式 Web 客户端进行编写。为了在前端实现对于生产设备的同步仿真,使用 3D 引擎 three.js 在页面中嵌入机器人设备的 3D 模型。考虑到前后端的快速通信需求,采用 GoLang 语言和 Beego 框架进行后台功能开发。为了实现对现场画面实时监控,在后台部署基于快速 UDP 互联网连接(quick UDP Internet connection, QUIC)协议的实时消息传输协议(real time messaging protocol, RTMP)直播服务器,接收视频推流并在前端播放。为了实现对“端”层设备的控制以及数据交互操作,集成 ModBus 协议通讯进行远程交互,封装 MoveJ 和 SpeedJ 指令控制机械臂运动,在进程间采用 Redis 数据库对数据进行缓存,采用 MySQL 数据库对数据持久化存储。为了能够实现系统设备在“云”与“边”层的分布式部署,并实现进程间的低延时通信调度功能,采用远程过程调用(remote procedure call, RPC)分布式框架实现系统不同层级进程间通讯的功能,并在通信协议上采用 QUIC 协议代替 TCP 协议进行数据传输,降低通信延时以增加系统的实时性。为了确保生产环境的人机安全,通过向部署的 YOLO 模型中输入生产环境图像,对机器人

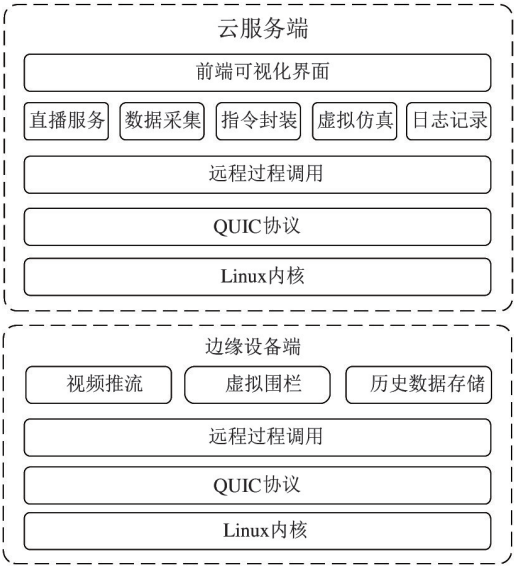


图 1 系统总体框架图

和工作人员进行识别框定,并根据空间位置关系生成警报信息和机器人控制决策。为了增加监控系统的及时性和安全性,将上述监测计算和控制决策功能从“云”层的云服务器移到“边”层的嵌入式设备上计算。

以分布式调度框架实现云端系统和边缘设备的协同工作,实现以下功能:(1)实现对多类工业机器人设备的远程连接、数据采集和运行控制功能;(2)针对工业生产环境中的机器人运行安全以及人机安全问题进行监控,及时进行异常警报和决策控制;(3)实现对不同设备、进程和语言的功能模块进行低延时通信和分布式调度;(4)为用户提供可视化前端,包括直观的监控界面和简洁操作面板。

2 系统硬件设计

2.1 系统硬件总体方案设计

本系统硬件总体架构如图2所示,系统以“云-边-端”三层构成。“云”层为高带宽大存储容量的云服务主机,向上接受用户的指令操作并可视化监控信息,向下进行功能调度和数据采集;“边”层为带算力的嵌入式设备,作为内网设备的工业网关与外网通信,同时采集摄像头数据进行实时计算监控;“端”层由部署在工业生产环境内网的一系列工业机器人以及监控摄像头组成。

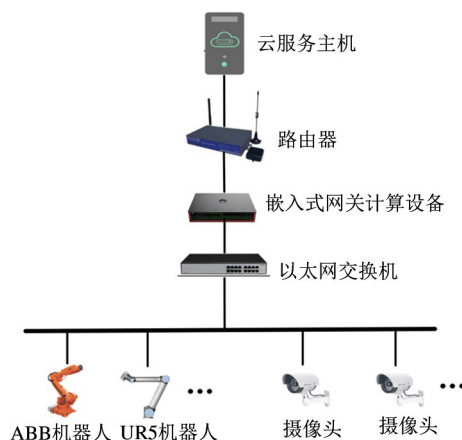


图2 硬件总体架构图

近年来科技高速发展,设备的存储和计算能力都得到了迅速的进步,以带算力的嵌入式设备代替

云计算主要有以下几个优点:(1)将计算和存储由边缘设备执行,能够降低云端的负荷,提升系统整体的计算速度,降低数据传输导致的网络延迟;(2)将数据在边缘端处理,同时将决策处理从云端移至边缘端,降低分布式拒绝服务(distributed denial of service, DDoS)恶意攻击的风险,增加系统的安全性;(3)增加系统的可扩展性,对系统进行扩展升级,只需通过增加或升级边缘设备,而不需要对云服务器进行资源升级;(4)降低云服务端故障、网络中断等意外因素带来的危害性,增加系统的可靠性。

2.2 系统硬件选型与分析

针对上述设计需求,选择TP-LINK TL-SG1024DT千兆网络交换机进行局域网组网;选用萤石C6C摄像头进行实时监控;嵌入式设备的选型需求为:作为网关设备需要有丰富的接口资源,同时要强劲的算力对监控画面进行实时检测。综合考虑算力、成本、硬件资源等因素,最终选择ATLAS 200计算卡结合扩展系统载板RTSO-7001进行系统构建,拓展载板如图3所示。利用系统载板RTSO-7001上多网口的优势,实现工业网关设备的功能;利用计算卡ATLAS 200的22 TOPS INT8算力以及最高16路30帧1080P视频的编解码能力的优势,作为局域网内的计算节点,对所采集的视频数据进行计算,实时输出监控结果。ATLAS 200与系统载板RTSO-7001结合作为“边”层的通信计算节点,将产生的监控数据实时上报至云端。而云端对各个嵌入式端节点发送来的数据信息进行汇总、处理、存储,并在前端界面进行可视化显示。

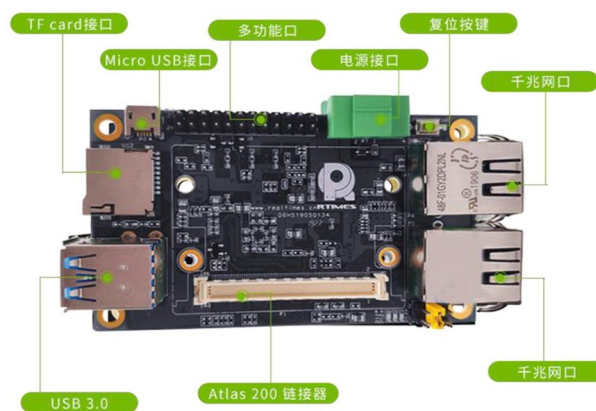


图3 扩展载板

3 系统软件设计

3.1 系统软件总体方案设计

本系统采用“边-云”协同的分布式网络结构对系统软件进行设计,依照功能逻辑将云端分为节点通信模块、远程控制模块、可视化监控模块和数据采集模块,将边缘节点分为节点通信模块、实时监控模块和决策控制模块,功能结构如图 4 所示。云端监控中心通过节点通信模块与远端的边缘计算节点通信,实现数据的交互以及任务的分布式调度。在此基础上通过云端以及边缘节点上的监控模块协同工作实现对生产环境和设备运行状况的可视化监控;通过云端以及边缘节点上的控制模块协同工作,实现对生产环境中机器人设备的远程调试功能,进而实现整个监控系统。

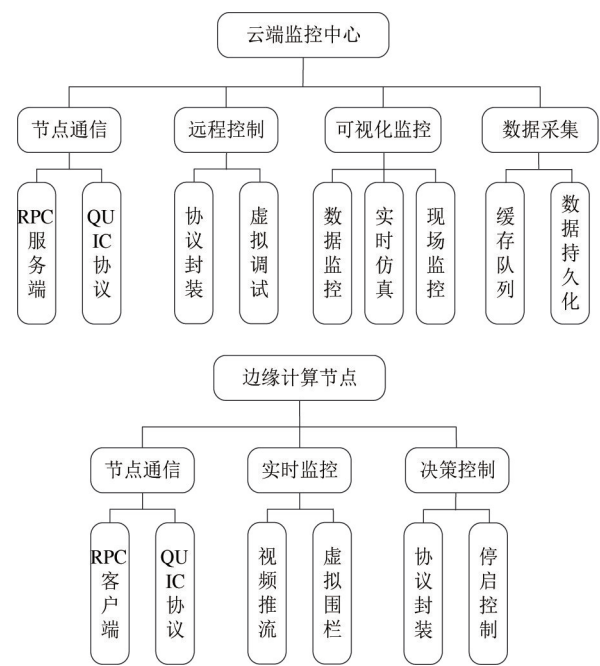


图 4 功能结构图

基于上述“边-云”协同的分布式结构设计思想,设计如图 5 所示的软件系统框架,包括云端监控中心以及边缘计算节点 2 个层级。以 2 个层级协同工作的模式,将系统以核心服务为划分,抽离出节点通信子系统、实时监控子系统和远程控制子系统,实现对远程工业机器人的监控。

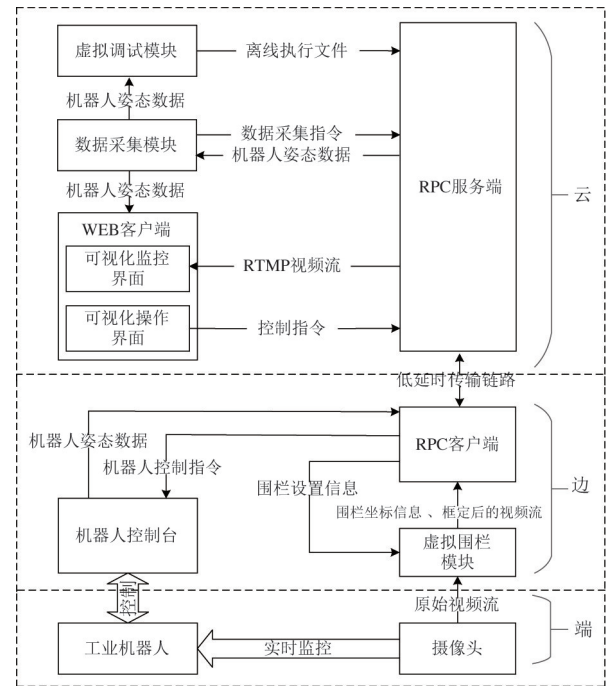


图 5 软件系统框架图

3.2 节点通信子系统

本文中的节点通信子系统采用 GoLang 语言进行开发,为了实现在不同语言、不同设备运行的进程间互相通信,设计 RPC 服务端和客户端进行任务的分布式调度。为了减少业务在不同网络环境下的网络耗时,在传输层采用 QUIC 协议代替传统的 TCP 协议进行数据传输。

3.2.1 协议选择

在通信协议的选择上,传统传输层协议包括 TCP、UDP 等协议,TCP 协议面向连接,强调的是传输的可靠性,但连接开销大、效率低。UDP 协议是无连接方式的协议,效率高速度快,但是传输不可靠。基于监控系统实时性和稳定性的设计需求,采用 QUIC 协议作为通信协议进行系统开发。QUIC 协议由 Google 定制,基于 UDP 实现,其集成了 TCP 协议和 UDP 协议的优势,可以在保证稳定传输的同时有效减少网络延迟^[13]。在本文中采用 QUIC 协议代替传统协议进行系统开发有以下优点:(1)传统协议内置于系统内核中实现,而 QUIC 基于客户端实现,在后续升级中,无需进行操作系统层级的修改,方便进行系统的迭代更新;(2)在协议性能上,与 TCP 协议相比 QUIC 采用多路复用设计,可以通

过一个连接同时进行多个请求,减少时间和连接资源的浪费;采用 TCP Fast Opemd 的设计,在已建立过连接的情况下无需握手就可以再次建立连接,减少等待时延;在重传机制上采用前向纠错(forward error correction)机制恢复传输中丢失的数据,减少重传造成的阻塞;对于拥塞控制、数据加密等机制也做了进一步的优化。使用 QUIC 协议可以保证系统在稳定安全地传输数据的同时,极大地提升访问速度增加系统的实时性。基于在 Web 端实时显示直播现场画面的监控需求,修改视频推流协议 RTMP 的传输层协议以提升系统监控性能。QUIC 协议在网络传输中的层级关系如图 6 所示,将 RTMP 的底层实现协议从传统的 TCP 协议修改成 QUIC 协议,达到降低推流卡顿的效果。

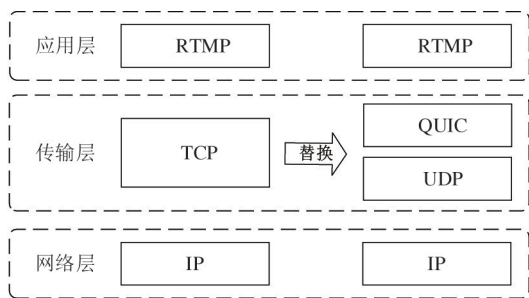


图 6 QUIC 协议层级关系图

3.2.2 RPC 系统设计

基于本文的应用场景,需要在云端远程调用 C++ 接口的目标检测推理引擎,实现对现场图像的实时检测。因此所实现的 RPC 功能需要在跨设备的同时实现跨语言的服务调度,而 GoLang 语言标准库提供的 net/rpc 包所实现 RPC 框架不能实现跨语言服务调用,同时底层协议的实现也已经封装好难以修改,不能满足设计需求。因此在标准库的基础上进行以下扩展设计。

(1)为了解决通讯问题,以 QUIC 协议为基础,建立客户端与服务端间的通信链路。

(2)为了解决服务寻址问题,为通信进程双方绑定固定的 IP 地址和端口号解决设备寻址问题;为服务方法绑定唯一的 UUID 号,并将 UUID 号注册至服务列表中,实现服务的寻址。

(3)为了实现跨语言的数据交互,在两端进程

中维护相同的 JSON 结构用于封装通信的数据。同时由于在底层网络协议中数据以二进制数据的形式传递,需要将传递的数据进行序列化,因此采用封装了通用编码协议的 GO 标准库" net/rpc/jsonrpc" 包对 JSON 数据进行序列化和反序列化操作。

上述 RPC 设计实现后端进程间跨语言、跨内存、跨设备服务的相互调用。RPC 进程调用流程如图 7 所示,在此基础上即可实现云端与边缘设备的协同工作。

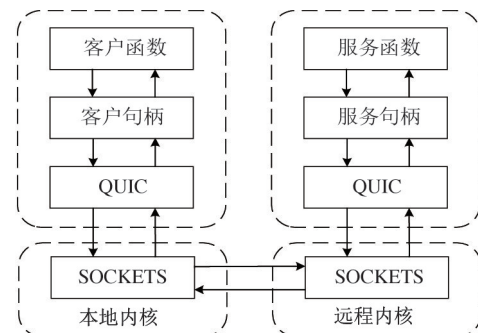


图 7 RPC 进程调用流程图

3.3 实时监控子系统

以“边-云”协同的模式实现监控子系统,将计算工作放在边缘设备上极大减少了数据上下行负担和云端的资源消耗。以前后端分离的形式开发 Web 系统,基于 Beego 框架实现前后端的通信,用户能随时随地通过浏览器在可视化的前端界面中,对设备的运行状态和人机安全进行远程监控。

3.3.1 数据采集模块

实现数据采集模块对生产环境中的工业机器人控制台进行数据读取、缓存和持久化存储,作为后续模块实时监控分析的数据支撑。本文设计数据采集模块采集机器人姿态数据,通过 RPC 框架在云端远程调用边缘设备上封装的数据采集协议,对 UR5 机器人和 ABB 机器人的控制台进行数据采集,并将数据上传至部署于云端的数据库中。

为了实现对多类工业机器人数据采集的通用化,首先在边缘设备上通过 Socket 连接实现数据采集模块与机器人控制台的数据通信,并以 ModBus 协议对工业机器人的数据进行采集;对不同机器人的寄存器地址进行封装注册,实现多个寄存器的读取。由于部分机器人服务器端口不支持多连接的问题

题,采用长连接和短连接的机制,对于只支持单连接的服务器端口在每次读写完毕后就断开连接;对于支持多连接的服务器端口,维护长连接进行通信以增加读写速度。对于不同的机器人封装不同的采集协议,对 ModBus 协议返回的报文信息进行解析转换,以便获取正确的数据,数据采集流程图如图 8 所示。

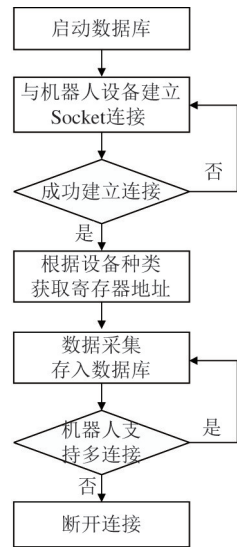


图 8 数据采集流程图

采用非关系型数据库 Redis 对读取到的数据进行缓存,以异步存取的方式,实现高速的数据采集、读取通路,以提升系统整体的运行效率,同时为后续模块提供交互物理信号。利用关系型数据库 MySQL 对设备的历史数据进行存储,允许用户根据时间段对历史数据查询,进而对设备运行历史状态进行分析。

3.3.2 虚实协同监控模块

为了更好地实现对生产环境中工业机器人运行状态的监控,设计了虚实协同的监控模块。以传统数据监控和视频监控模式为基础,加入对工业机器人的实时仿真。以现实物理设备数据实时驱动仿真环境中的虚拟 3D 模型运动,用户能在仿真环境中拖动视角对运行设备在不同角度进行观测,增加设备监控的维度。

如图 9 所示,设计前端界面对设备运行状态的监控信息可视化输出,包括左侧的视频监控模块、右上角的数据监控模块和右下角的 3D 仿真模型。

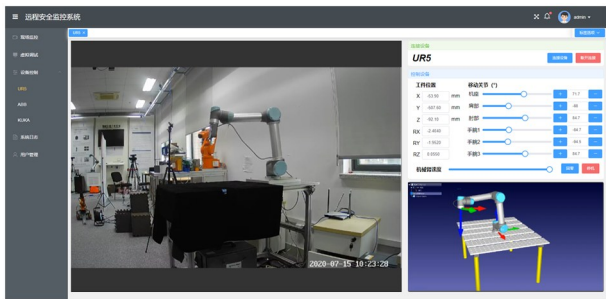


图 9 虚实协同监控前端界面

为了实现对远程监控画面的直播功能,在云端部署直播服务器,用于接受由边缘设备上推流来的视频流数据。在前端页面中嵌入 flv.js 插件,对直播服务器上的视频数据进行拉流,输出视频画面。为了实现对数据的实时监控,从 Redis 数据库中读取实时数据,并设计数据可视化框对数据进行展示。

为了能实时驱动前端的机器人模型,使用 3D 引擎 three.js 在页面中嵌入机器人设备的 3D 模型实时仿真。在后端读取 Redis 数据库中的实时设备数据,并通过 Beego 框架传输到前端写入虚拟模型的数据标签中,驱动虚拟模型运动。在设备故障时,用户可以根据时间信息从 MySQL 数据库中导出历史数据,驱动模型进行轨迹复现,并在运行时通过拖动模型的视角在不同角度观察设备的运行状态,帮助进行故障的分析。

在监控系统的并发性能上,考虑到服务器、网络和设备带宽上限,视频传输会成为系统并发数量的瓶颈,因此本系统将视频数据与设备运行数据分线程传输。对于视频数据,由前端播放器界面的开关控制系统视频推流的停启,限制同一时刻最高传输 4 路 720P 的视频监控画面,以保持系统数据传输的稳定畅通;对于设备运行数据,系统会实时采集所有与系统建立连接的设备的运行数据,并缓存于 redis 中等待前端页面读取。通过上述机制保障系统对于多机器人设备的并发监控性能。

3.3.3 虚拟围栏模块

为了在生产环境中进行实时人机安全监控,设计虚拟围栏模块,通过分布式调度方法实现“边-云”协同的监控模式。在云端远程调用运行在生产环境侧边缘计算设备上的推理引擎,实时检测双视角下的生产环境视频数据,进而推理出人机的位置关系,

并将检测结果结合视频数据推流至云端实时显示,前端界面如图 10 所示。

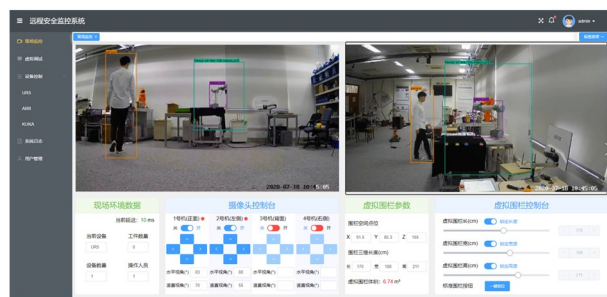


图 10 虚拟围栏前端界面

虚拟围栏如字面意义所示,在运行的机器人周围实时框定一个 3D 的工作空间,当检测到有人进入虚拟围栏的空间范围内时,触发报警并在距离过近时对机械臂下发急停指令以确保人机安全。将检测结果以 RTMP 推流的形式实时推送至云端直播服务器中播放,实现生产环境下的人机安全监控。

基于文献[14]的目标检测算法训练了目标检测模型对双视角下的人员和机器人进行目标识别和框定,并在此基础上生成机械臂周围的虚拟围栏,同时利用 DeepSort 算法对目标进行跟踪,解决遮挡问题。首先拍摄不同角度姿态、远近和明暗的机器人设备和人物图像共 4800 张建立数据集,每类检测对象的图片数量为 1600 张,在标定后将数据集分成 80% 的训练集和 20% 的测试集进行训练。将训练后的模型利用华为的开发工具链 Mindstudio 转换为 Atlas 200 计算卡上能支持的 om 格式离线模型。利用 ACL 库抽象出的 C++ 功能接口实现推理引擎的开发,进而实现对原始图像数据的预处理、推理模型的调用、推理结果的后处理操作。

通过双路摄像头从 2 个垂直的视角拍摄运行中的机器人,并将两路图片帧送入推理引擎中即可根据运行机器人的中心点生成 3D 虚拟围栏,同时可以推理出人员是否进入机器人的工作空间中。

3.4 远程调试子系统

为了方便用户对设备的远程控制,设计远程调试子系统,提供角度控制和虚拟调试 2 种远程控制方式。设计前端交互控制台供用户操作,控制台界面如图 11 所示。用户可以在控制台上以按键远程

连接机器人、控制机器人停启,拖动滑块来控制工业机器人的关节角和运动速度。

为了实现对机器人的关节控制,首先与机器人控制器建立 Socket 通信连接,在此基础上将机器人脚本命令封装成 MoveJ 和 SpeedJ 指令,控制机器人实现不同的运行效果。其中 MoveJ 指令控制机器人各个关节的角度值,SpeedJ 指令控制机器人各个关节的速度值。



图 11 控制台界面

考虑到在 Web 端远程调试机器人时,不可避免会因数据传递延迟问题导致操作延迟,在无监督的情况下存在过调甚至损坏机器人的风险,因此基于虚拟调试技术为系统加入虚拟调试功能。虚拟调试技术是在虚拟环境中调试控制设备的代码,通过虚拟仿真测试和验证系统方案的可行性,再将调试代码应用到真实的场景中的技术^[15]。

通过 VNC 协议实现远程控制部署于现场的调试主机,在调试主机上运行 RoboDK 软件构建的生产环境虚拟仿真模型。在虚拟环境中进行调试,将调试完毕的机器人运动轨迹生成机器人离线程序,通过安全文件传输协议(secure file transfer protocol, SFTP)传送到机器人服务器中执行,完成对机器人程序的虚拟调试功能。从而通过浏览器就可对设备问题、运行方案等进行虚拟调试实验,虚拟调试界面如图 12 所示。

4 实验验证

基于本文所设计的远程监控系统构建了实验平

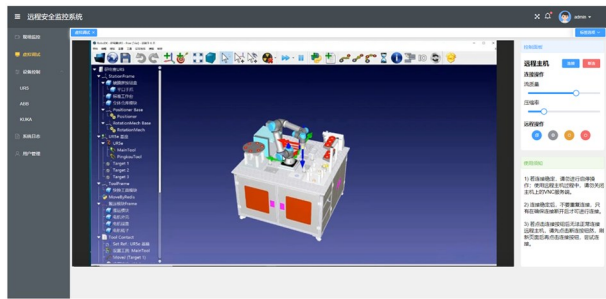


图 12 虚拟调试界面

台。实验平台包括 1 张华为 ATLAS 200 计算卡、1 张 RTSO-7001 拓展载板、2 个萤石 C6C 网络摄像头、1 台路由器、1 台千兆交换机、1 台 UR5 机器人及其控制台、1 个阿里云服务器,以及 1 台笔记本电脑和若干网线。

4.1 设备远程连接实验

以 UR5 机器人作为监控对象,将边缘设备的 WAN 口通过网线接路由器连接外网。将边缘计算设备的 LAN 口、机器人控制器网口和摄像头通过网线连接在同一交换机上,并配置 IP 地址在同一网段下。在建立连接后通过电脑浏览器访问部署在云端的 Web 系统并登录,结果如图 13 所示。通过页面上的连接按钮连接远程设备,前端获取到了现场画面,并远程连接到了机器人控制器中获取到了关节数据。

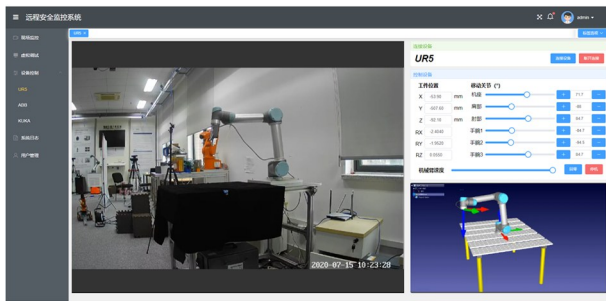
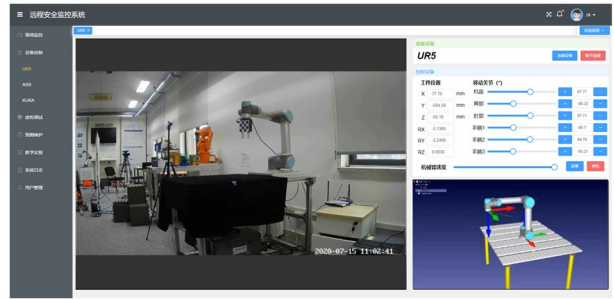


图 13 设备连接控制界面

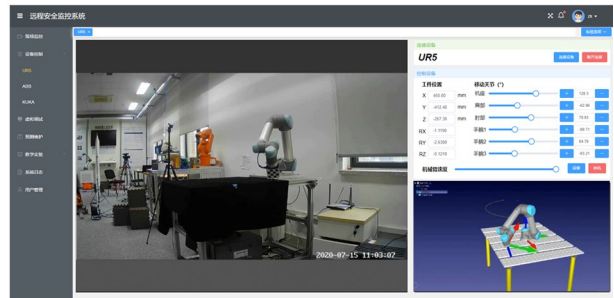
4.2 远程监控实验

通过 Web 端监控界对机器人运行状态进行监控。在连接设备后,左侧监控开启获取实时监控画面,右侧上半部分控制台中获取到当前机器人姿态的关节数据,右侧下半部的仿真环境可以看到当前机械臂处于的实时位姿,如图 14(a)所示。通过控

制台对机器人关节角进行调整,可以看到刀具位置、关节角度等数据随机械臂运动实时更新,同时 仿真环境中模型的姿态随着实际机器人的变化而变化,结果如图 14(b)所示。



(a) 设备初始位置



(b) 设备控制后位置

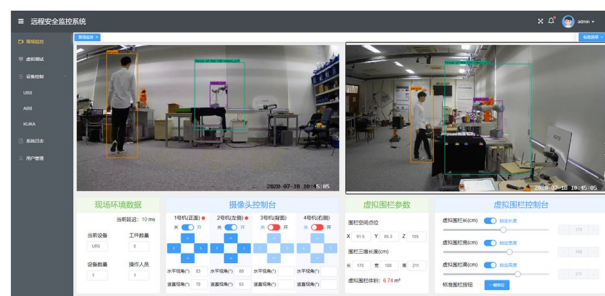
图 14 运行状态监控实验结果

在 Web 端监控界面对虚拟围栏效果进行验证。在连接设备后,界面左右两侧获取到处理后的视频流数据,并对画面中的机器人和人进行实时检测框定,如图 15(a)所示,此时测试人员在虚拟围栏范围外。当测试人员进入虚拟围栏的范围内时,前端产生报警信息,并触发机械臂的急停措施,结果如图 15(b)所示。

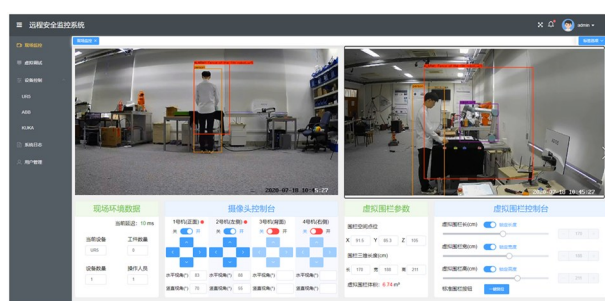
4.3 系统实时性实验

以本文系统作为测试平台,以 UR5 机器人作为目标连接对象,在 4G 网络环境下进行实时性测试,所测试的性能指标包括数据采集时间、指令响应时间和视频传输质量测试。测试方式如下。

基于文献[16]的方法,以 WebSocket 协议为基础实现双工数据传输机制对数据进行传输,实现所有功能模块部署在一起的单一应用架构监控系统。将该系统与本文系统在平均数据采集时间和平均指令响应时间上进行对比测试。程序控制以30ms的



(a) 未触发虚拟围栏



(b) 触发虚拟围栏

图 15 虚拟围栏监控实验结果

时间间隔产生数据采集指令对机器人关节角度采集,共计 3000 次计算平均数据采集时间。用程序控制以 2 s 一次的频率产生角度控制指令,控制机器人以相同的速度和角度前后摆动,共计 3000 次,计算出平均指令响应时间。比较结果如表 1 所示,由表 1 可知,以 QUIC 协议替代 TCP 协议,同时以分布式服务框架代替单一应用框架可以为远程监控系统提升 20% 左右的实时性。

表 1 数据采集和指令响应实时性对比

性能指标	对比系统	本文系统
平均数据采集时间	36.3 ms	26.7 ms
平均指令响应时间	89.1 ms	68.3 ms

基于文献[17]的方法,搭建 TCP 为协议基础的流媒体服务器实现直播测试,将该系统与本文系统进行视频传输质量测试对比。以 1080P 的画质采集 2 路监控视频,进行处理和推流操作 15 min。分别在画面采集到和在云端获取到处理后的图片帧时记录时间戳,以时间戳相减获取监控延时,并记录缓存中图片帧的堆积数,当超过 10 帧时记录为一次播放卡顿。比较结果如表 2 所示,由表 2 可知以 QUIC

协议替代 TCP 协议,可以在画面直播时减少延时和卡顿,提高了系统的实时性。

表 2 系统实时性对比

性能指标	对比平台	本文系统
平均监控延时	1513.7 ms	1185.3 ms
视频卡顿计数	37 次	6 次

通过以上实验可知,该系统可以通过实时图像处理 and 实时模型驱动对设备运行状态和人机安全进行监控;通过对多种协议的封装实现对远程设备的调试;通过 QUIC 协议可以减少网络延时,同时结合分布式框架可以减少协议的封装、增加代码的执行效率增加系统的实时性。该系统运行稳定,可以高效地对工业机器人生产环境中的多类安全问题进行监控,有效地提升生产环境中的安全系数。

5 结论

本文设计并实现了虚实结合的低延时工业机器人监控调试系统。设计了 B/S 架构的客户端系统,为用户提供了直观的可视化界面,能够在只有一个浏览器的情况下对生产现场和机器人进行远程监测与控制。通过在前端界面嵌入虚拟仿真环境,对设备运行状态进行多维度的监控。通过目标检测算法为机器人加装虚拟围栏,能有效降低人机安全事故发生的风险。而基于数字孪生的虚拟调试技术,能在机器人发生故障后,通过虚拟仿真找出问题根源,提高效率的同时降低成本。整套系统通过 RPC 框架结合 QUIC 协议实现了高效稳定的分布式调度,实现了较好的实时性。实验结果表明所设计的虚实结合的低延时工业机器人监控调试系统运行效果良好,在面对工业生产环境中机器人的安全问题时,已经具备相当程度的实际作用,为工业生产环境中机器人提供了一种合适的方案。

参考文献

- [1] 战戟. 工业 4.0 从自动生产到智能制造[J]. 智库时代, 2020, 226(6): 3-4
- [2] 王喜文. 工业 4.0、互联网+、中国制造 2025 中国制造业转型升级的未来方向[J]. 国家治理, 2015(23):

- 12-19
- [3] 贾计东, 张明路. 人机安全交互技术研究进展及发展趋势[J]. 机械工程学报, 2020, 56(3):16-30
 - [4] 邢琳. 工业机器人安全之生命周期风险评估[J]. 安全与电磁兼容, 2019(4):97-100
 - [5] 潘力, 高伟强, 刘建群, 等. 基于云计算的喷涂机器人远程监控诊断系统的研究[J]. 组合机床与自动化加工技术, 2019, 550(12):52-56
 - [6] PAN J, ZI B, WANG Z, et al. Real-time dynamic monitoring of a multi-robot cooperative spraying system[C]//IEEE International Conference on Mechatronics and Automation, Tianjin, China, 2019:862-867
 - [7] 孙恺廷, 朱隽垚, 于存贵, 等. 工业机器人三维虚拟监控系统的设计与实现[J]. 机械制造与自动化, 2020, 49(5):154-156, 210
 - [8] VOGEL C, FRITZSCHE M, ELKMANN N. Safe human-robot cooperation with high-payload robots in industrial applications[C]//ACM/IEEE International Conference on Human-robot Interaction, Christchurch, New Zealand, 2016:529-530
 - [9] 肖鸣杰. 基于 ARM 的工业远程监控系统设计[J]. 自动化技术与应用, 2019, 38(6):74-77, 88
 - [10] ATMOKO R A, YANG D. Online monitoring&controlling industrial arm robot using MQTT protocol[C]//IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems, Guilin, China, 2018:12-16
 - [11] 李瑾博. 基于 TCP/IP 协议的工业控制网络远程数据通信网关[J]. 电子技术与软件工程, 2017(17):13
 - [12] 曲楠, 宋万强. 基于以太网的机器人在线监控系统设计[J]. 工业控制计算机, 2020, 33(11):10-12
 - [13] 李学兵, 陈阳, 周孟莹, 等. 互联网数据传输协议 QUIC 研究综述[J]. 计算机研究与发展, 2020, 57(9):1864-1876
 - [14] 张富凯, 杨峰, 李策. 基于改进 YOLOv3 的快速车辆检测方法[J]. 计算机工程与应用, 2019, 55(2):12-20
 - [15] LECHLER T, FISCHER E, METZNER M, et al. Virtual commissioning-scientific review and exploratory use cases in advanced production systems[J]. *Procedia CIRP*, 2019, 81:1125-1130
 - [16] 张文, 牟艳, 高振兴, 等. WebSocket 技术在工业设备数据实时监测中的应用[J]. 计算机与现代化, 2017(11):111-115
 - [17] 李志瑞. 高性能移动直播场景下 QUIC 协议研究与应用[D]. 广州:华南理工大学计算机科学与工程学院, 2018:52-79

Monitoring and debugging system for low delay industrial robot combining virtual with real

YU Xinyi, YING Haozhe, ZHENG Kun, WANG Ben, OU Linlin

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

An monitoring and debugging system for low delay industrial robot combining virtual with real is developed to solve the problems of single monitoring means, incomplete information, difficult troubleshooting and high delay in monitoring system. Firstly, a distributed monitoring system of cloud-edge-device synergy is implemented based on remote procedure call (RPC) framework and quick UDP Internet connection (QUIC) protocol, which can call resources across languages and servers and reduce system latency. A Web monitoring system with good human-computer interaction is developed based on the mode of Beego framework and Vue front-end separation. Encapsulate communication protocols for multi-class robots and interfaces of various databases in the back-end, realize data collection and storage of robots. The virtual model of the front-end is driven by real-time data collected to monitor the running status of devices in the mode of virtual-real combination. In man-machine security, based on the target detection algorithm YOLO a three-dimensional workspace is delimited for the robot in the form of a virtual fence, and the spatial location relationship between humans and robots is detected in real time to ensure human safety. In order to ensure the security of remote control, the virtual debugging technology is used to pre-simulate movements of robots in the virtual environment. After verifying the security, offline files are generated and sent to the robot for execution to ensure the security of remote debugging. Finally, the feasibility of the system is verified by experiments.

Key words: cloud-edge-device synergy, remote monitoring, virtual-real combination, quick UDP Internet connection (QUIC) protocol, virtual fence