

基于阿里云的裁断机监控系统设计

沈鑫睿, 杨冰冰, 徐建明, 何德峰

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 采用云技术和 WEB 技术, 设计了一款基于阿里云的裁断机监控系统, 系统主要分为阿里云 WEB 监控端和本地运动控制端, 本地运动控制端主要工作是采用 Winform 框架在本地服务器搭建了上位机系统, 通过 OPC UA 协议与裁断机控制器进行数据交互, 控制裁断机完成对冲裁料的加工; 阿里云 WEB 监控端采用前后端分离的模式进行开发, 基于 VUE 框架设计了前端用户操作界面, 并通过 Canvas 标签搭建了排样结果可视化区域; 基于 Django 框架设计后端, 主要编写了刀模排样算法用于生成单一刀模样片在冲裁料上的排样方案, 并将开发完成的系统部署至阿里云服务器; 最后, 通过搭建 MQTT 数据交互模块实现裁断机冲裁命令的下发和运行数据的回传。

关键词: 云技术; WEB 监控; 图形排样; MQTT 协议; 裁断机

Design of Monitoring System for Cutting Machines Based on Ali Cloud

Shen Xinrui, Yang Bingbing, Xu Jianming, He Defeng

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

Abstract: Adopting cloud technology and Web technology, a cutting machine monitoring system is designed based on Ali Cloud. The system is mainly divided into Ali Web monitoring terminal and local motion control terminal. The upper machine on local server system is built by Winform framework. And the server system through OPC UA protocol to communicate with cutting machine controller, which can control cutting machine to process cutting material. Ali Cloud Web monitoring terminal is developed by the mode of separating front end and back end. The front-end user operation interface is designed by VUE framework. In the operation interface, layout result visualization area is built by Canvas tag. The back-end is designed based on Django framework, which mainly design the cutter layout algorithm for generating a single cutter shape slice layout scheme on blanking material, then deploy the developed system to Ali Cloud server. Finally, the issuing of control commands and the return of running data are realized by the MQTT data interaction module.

Keywords: cloud technology; Web monitoring; graphical layout; MQTT protocol; cutting machine

0 引言

随着智能制造时代的来临, 在工业生产领域提出了工业物联网^[1]的概念, 其目的是通过工业资源的网络互联、数据互通和系统相互操作, 实现制造过程的合理优化和制造环境的快速适应, 以达到资源的高效利用。同时, 云平台技术的提出使得互联网真正进入了云计算时代, 依托云平台将本地软、硬件资源虚拟化, 可以改变工业生产的模式并提升效率。

皮革加工行业中, 裁断机作为一种被广泛使用的传统手工设备, 加工企业主要依靠人工方式进行裁断作业, 该方式不仅需要大量劳动力, 且对于冲裁料的利用率低下, 难以达到理想的效果。为了在一个有限的冲裁区间内实现对冲裁样片的充分排列, 提高冲裁料的利用率, 许继影^[2]等人提出基于两阶段排放算法的矩形排样优化方法、Satoshi Fujita^[3]等人研究了可旋转物品在二维装箱问题中的

解。如今对于裁断机等传统工业设备的联网研究多数只是对其运行数据进行监测^[4], 如孙恺廷^[5]等人设计的工业机器人三维虚拟监控系统、胡晓轩^[6]等人面向船舶制造车间的数据采集与监控系统, 这些研究往往是依靠各种传感器设备将设备运行数据进行采集, 之后通过对采集的数据做进一步分析, 得出设备是否运行正常等结论, 但并未改变设备的控制模式。目前, 裁断机的控制系统大多采用客户端/服务端模式搭建, 并嵌入了排样算法实现裁床对冲裁料的自动加工。该模式下, 对于每个机床终端在本地均需要为其配备一套相关设备^[7], 并安装客户端软件, 当终端数量增加时, 其配套设备的成本会十分巨大。

针对上述问题, 本文在裁断机原有的客户端/服务端控制模式基础上, 结合云技术和 WEB 技术设计了一款基于阿里云的裁断机监控系统, 依靠云平台减少了系统对本地硬件资源的需求。系统主要分为本地运动控制端和阿里云 WEB 监控端。其中, WEB 监控端采用前后端分离的方式

收稿日期: 2020-10-16; 修回日期: 2020-11-16。

作者简介: 沈鑫睿(1995-), 男, 浙江宁波人, 硕士研究生, 主要从事工业物联网技术方向的研究。

何德峰(1979-), 男, 浙江义乌人, 教授, 博士生导师, 主要从事模型预测控制理论与应用方向的研究。

通讯作者: 徐建明(1970-), 男, 江西吉安人, 教授, 博士研究生导师, 主要从事工业机器人和运动控制技术方向的研究。

引用格式: 沈鑫睿, 杨冰冰, 徐建明, 等. 基于阿里云的裁断机监控系统设计[J]. 计算机测量与控制, 2021, 29(5): 107-111.

搭建, 前端用户操作界面基于 VUE 框架编写, 后端基于 Django 框架编写。本地运动控制端的工作主要是在本地服务器搭建了上位机系统, 采用 OPC UA 协议实现了与裁断机设备控制器的信息交互, 采用 MQTT 协议完成了与云端的信息交互。最终使得用户可以通过访问部署在阿里云的操作界面完成对本地裁断机设备的监控。

1 裁断机监控系统设计方案

1.1 系统硬件结构框架

系统硬件结构如图 1 所示。本地运动控制端由服务器、控制器和裁断机设备组成。服务器搭建有上位机系统, 该上位机系统通过 OPC UA 协议^[8]实现一台服务器与多个控制器的数据交互, 控制器采用 AM335X 芯片并搭建 CodeSys 系统, 通过 EtherCAT 总线实现对裁断机设备的控制。WEB 监控端则部署在阿里云服务器上, 实现用户能够通过访问相关的网站地址, 对本地的设备完成监控。



图 1 系统硬件结构图

1.2 系统软件结构框架

整个监控系统以基于 CoDeSys^[9]开发的裁断机控制系统为研究对象进行搭建, 主要完成阿里云 WEB 监控端和本地上位机系统的开发, 实现对裁断机的监控, 整体软件设计框图如图 2 所示。

通过在本地服务器搭建上位机系统, 实现云端与本地控制器的数据交互。上位机主要采用 Winform 框架开发, 通过嵌入 MQTT 客户端^[10]和 MySQL 客户端^[11]实现与云端的数据通讯; 通过嵌入 OPC UA 客户端实现与控制器的数据通讯。其中 MQTT 主要完成一些实时控制命令的下发; MySQL 用于存储设备的运行数据; OPC UA 实现将裁断机运行数据下发至设备控制器以及回传设备当前的运行状态信息。

阿里云 WEB 监控端分为前后端两部分, 前端采用 VUE 框架开发, 用户可以操作前端页面生成相关控制命令并依靠嵌入的 MQTT 客户端完成命令的下发; 后端采用 Django 框架^[12]搭建, 编写了排样算法负责生成裁断机设备的运行数据, 并保存至数据库。

系统整体的监控流程如下: 用户使用平板电脑等移动

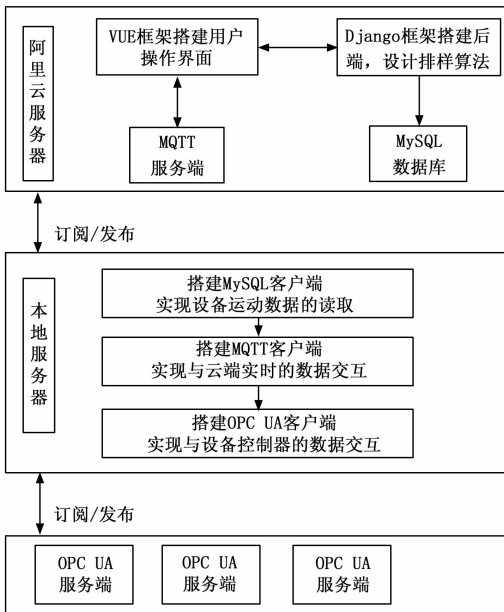


图 2 系统软件结构图

终端访问部署在云服务器的 WEB 界面, 生成合适的刀模排样结果^[13-14]并下发控制命令至本地服务器, 本地服务器中的上位机监听到云端的控制命令后, 会去数据库中读取对应的冲裁数据并下发至目标裁断机的控制器, 控制器根据这些数据控制裁断机完成对冲裁料的加工。在冲裁的过程中, 本地服务器会实时返回冲裁过程中裁断机的运行状态, 如当前正在冲裁的样片序号、本次对冲裁料的加工是否完成等。

2 本地上位机系统

本地对于裁断机设备的控制系统主要采用 C/S (客户端/服务器端) 模式进行搭建, 主要由服务器与设备控制器组成。在服务器搭建有上位机系统并嵌入了 OPC UA、MQTT 以及 MySQL 客户端。其中, OPC UA 客户端用于和裁断机控制器实现信息的交互; MQTT 客户端模块实现与云端实时数据的传输; MySQL 客户端模块实现对云端 MySQL 数据库的存取操作。

2.1 基于 OPC UA Client 的数据交互模块

传统的工业自动化解方案由于在设备间的通信上采用不兼容和不可互操作的差别化标准, 因而存在数据交互的困难。OPC UA 协议采用客户端/服务器模式和发布者/订阅者模式为不同设备间实现信息的交互提供了方法。通过在地址空间定义节点类并实例化, 采用层次结构进行访问简化了客户端的访问。C# 语言可以实现 OPC UA 的自动化接口, 开放给用户一些方便的接口, 支持自动配置、过程控制和数据存取功能的开发。通过在窗口界面中搭建用户输入和响应事件模块, 实现了浏览 OPC UA 服务器、修改数据以及订阅数据等功能, 其流程图如图 3 所示, 具体实现方式如下:

(1) 访问 OPC UA 服务器节点: 在主程序中引用

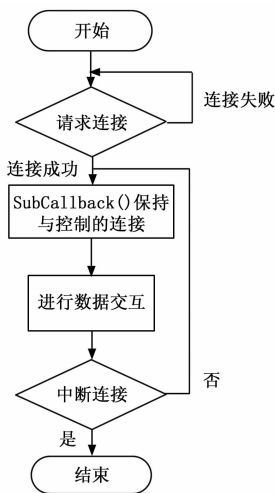


图 3 OPC UA 协议数据交互流程图

“OpcUaHelper. Forms”库, 使用库中的 FormBrowseServer () 函数实例化一个 MyForm 对象, 该对象能够以结构层次的方式对 OPC UA 服务器中的节点进行访问。在访问过程中, 能够浏览服务器所开放的相关节点信息。

(2) 订阅节点数据: 使用“OpcUaHelper. Forms”库中的 OpcUaClient () 函数实例化一个 OPC UA 客户端对象, 该对象内部包含了一系列对于服务器节点信息操作的方法。本系统采用其中的 ConnectServer () 方法实现对 OPC UA 服务器的连接, 采用 ReadNode () 方法实现对 OPC UA 服务器的设备运行数据的采集, 将所采集的数据转成字符串形式并用 C# 语言中的 String 变量进行保存, 用于后续与云端 MQTT 服务器进行数据交互。

(3) 修改节点数据: 采用“OpcUaHelper. Forms”库中的 WriteNode () 方法实现对 OPC UA 服务器相关数据的修改, 该函数均是通过访问 OPC UA 服务器中节点的各个层次结构, 在 TCP/IP 通讯协议的基础上完成对裁断机设备运行参数的修改。

2.2 基于云平台的数据交互模块

为了实现本地与云端的数据交互, 在上位机中嵌入了 MQTT 客户端模块和 MySQL 客户端模块。其中, MQTT 主要用于和云端的实时数据交互, 比如接收下发的运行设备编号和回传设备运行状态信息等, MySQL 主要存放生成的排样方案所对应的裁断机运行数据。

(1) MySQL 客户端模块: MySQL 是一个关系型数据库管理系统, 关系型数据库会将数据保存在不同的表中以增加存取速度并提高灵活性。如今, MySQL 已经为各种编程语言提供了 API 接口, 同时能够作为一个库嵌入其他的软件中, 使嵌入客户端的方式实现数据的存储得以实现, 不仅提高了数据交互的实时性, 也降低了本地客户端的内存空间消耗。本系统的数据读取模块采用 C# 语言进行搭建并嵌入到本地上位机系统, 在功能上实现对设备运行数据的读取和解析。

(2) MQTT 客户端模块: 目前基于 HTTP 协议的物联

网系统开发的方式被频繁采用, 但在一些嵌入式系统或网络带宽昂贵的开发情况下, HTTP 协议并不适用。MQTT 作为一种低开销、低带宽占用的即时通讯协议, 在工业物联网、小型设备等方面有广泛的应用。因此本客户端选择采用 MQTT 协议搭建数据交互模块, 实现对设备冲裁指令的下发和冲裁状态信息的回传。

3 阿里云 WEB 监控端

阿里云 WEB 监控端基于 B/S (浏览器/服务器) 架构进行搭建, 采用前后端分离的方式开发并部署至阿里云服务器。其中, 前端 WEB 操作界面基于 VUE 框架编辑, 开发了用户操作界面和排样结果可视化区域。后端采用 Django 框架搭建, 主要编写了刀模样片排样算法, 可以根据刀模图形的 DXF 文件信息^[15]生成裁断机设备的控制参数。在此基础上, 通过嵌入 MQTT 客户端模块实现和本地服务器的信息交互, 监控端功能框图如图 4 所示。

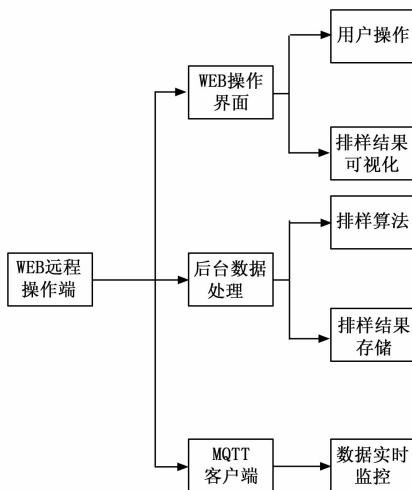


图 4 WEB 监控端功能框图

3.1 刀模排样算法

为了实现对冲裁料的充分利用, 需要对刀模样片进行排样。排样的目标是在特定区域的矩形冲裁料内放置尽可能多的样片, 且样片之间不能互相重合。本文根据实际冲裁情况设计了针对单一刀模样片的排样算法, 算法将冲裁方案主要分为水平垂直正排、水平对头单排垂直正排、水平正排垂直对头单排 3 种。

如图 5 所示, 将图中冲裁料坐标系记为 $\{B\}$, 冲裁料第 i 行、 j 列刀模样片的坐标系记为 $\{B_{ij}\}$, $\{B_{ij}\}$ 的原点相对于坐标系 $\{B\}$ 的坐标值为 x_{ij} 、 y_{ij} , 其旋转角度为 θ_{ij} 。令 $d_{ij}(\theta_{ij}, \theta_{i(j+1)})$ 为冲裁料上排列的第 i 行、 j 列样片与第 i 行、 $j+1$ 列样片的水平步距, d_{01} 为 $\{B_{11}\}$ 坐标原点到冲裁料 Y 轴的水平距离, $d_{N(N+1)}$ 为第 1 行、 N 列样片坐标系 $\{B_{1N}\}$ 的坐标原点到该样片右侧边缘切线的水平距离。 $h_{ij}(\theta_{ij}, \theta_{(i+1)j})$ 为在冲裁料排列的第 i 行、 j 列样片与第 $i+1$ 行、 j 列样片的垂直步距, h_{01} 为 $\{B_{11}\}$ 坐标原点到冲裁料 X 轴边界的水平步距, h_{34} 为第 3 行、1 列样片坐标系

$\{B_{31}\}$ 坐标原点到该样片上方边缘切线的垂直距离。

水平垂直正排方案的排列情况如图 5 所示, 在该方案下, 冲裁料上排列的样片旋转角度均一致。

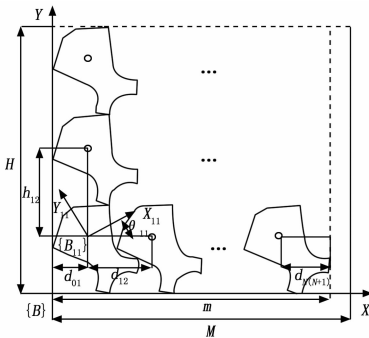


图 5 水平垂直正排示意图

水平对头单排垂直正排方案的排列情况如图 6 所示, 在该方案下, 冲裁料上排列的样片奇数列和偶数列的旋转角度差 180° 。

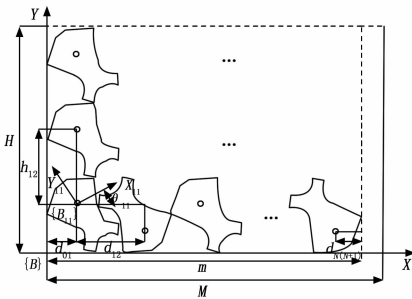


图 6 水平对头单排垂直正排示意图

水平正排垂直对头单排方案的排列情况如图 7 所示, 在该方案下, 冲裁料上排列的样片奇数行和偶数行的旋转角度差 180° 。

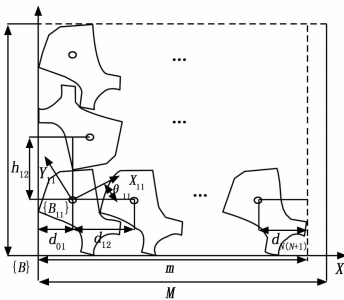


图 7 水平正排垂直对头单排示意图

记冲裁料一行能够排列的样片个数为 N , m 为一行样片的总宽度, M 为冲裁料的宽度, 有 $m = d_{01} + \dots + d_{(N-1)N}$, $m < M$ 。三行排样的总高度记为 H , $H = h_{01} + h_{12} + h_{23}$ 。排样方案对冲裁料的利用率记为 λ , 有 $\lambda = N / (M \times H)$ 。通过调整样片的旋转角度 θ_{ij} , 比较上述 3 种排样方案的 λ , 取 λ 为最大值所对应的排样方案为最终结果, 并将排样方案中所有样片坐标系原点相对冲裁料坐标系 $\{B\}$ 的坐标值 x_{ij} 、 y_{ij} 与样片的旋转角度 θ_{ij} 保存至数据库, 用于后续下发至裁断机设备控制器, 流程图如图 8 所示。

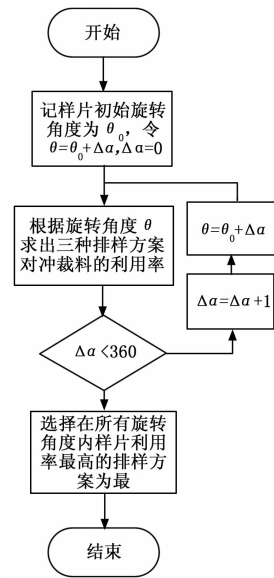


图 8 排样方案生成流程图

3.2 数据实时监控

为了能够实现云端对本地设备进行实时的数据交互, 本系统依靠 MQTT 协议在阿里云服务器搭建了实时监控平台, 实现了与 MQTT 服务器的数据交互、裁断机冲裁命令的下发以及冲裁过程状态信息回传等功能, 具体实现方式如下:

(1) 与 MQTT 服务器交互: 在前端 VUE 项目中引入 Paho 功能包, 实现 MQTT 客户端功能, 生成 MQTT 客户端前编辑用于连接 MQTT 服务器的参数列表, 如服务器 IP 地址、端口号等, 具体如下 `ServerUri='mq.tongxinmao.com'` `ServerPort=18832`; `TimeOut=5`; `KeepAlive=100`; `CleanSession=false`; `SSL=false`; 在此基础上调用 `new Paho.MQTT.Client` 函数, 创建 MQTT Client 实例, 编写 `onConnect()` 函数和 `onConnectionLost()` 函数用于登录和断开 MQTT 服务器; 编写 `WriteToStatus()` 函数, 用于描述 MQTT 状态; 编写 `onMessageArrived()` 函数, 用于接收 MQTT 服务器的消息, 实现与 MQTT 服务器数据交互功能。

(2) 冲裁命令的下发: 在前端 VUE 项目中编写 MQTT 客户端函数, 当用户点击发送按钮时, 将待运行的裁断机编号传至 MQTT 服务器中特定的主题文件, 本地服务器监听到该主题有数据会将主题中的数据读至本地并去云端数据库中取得运行数据, 完成下发。

(3) 冲裁过程状态信息实时回传: 当裁断机设备在运行时, 搭建在本地服务器会通过 MQTT 协议以一定的时间间隔将运行数据上传至云端, 该数据会被搭建在前端界面的 MQTT 客户端订阅并在可视化区域进行标记, 实现了冲裁过程中对裁断机运行状态的实时监控。

4 实验结果与分析

对于本文设计的控制系统, 通过实验验证其可行性。在 WEB 浏览器中输入云端 IP 地址访问操作界面, 在界面中选

用具有明显大小头的且形状不规则 1 号刀模, 点击 3 种排样方式最优解按钮, 在可视化区域可观察到 1 号刀模样片在一张宽为 1 400 mm 冲裁料上的排样结果, 如图 9 所示。

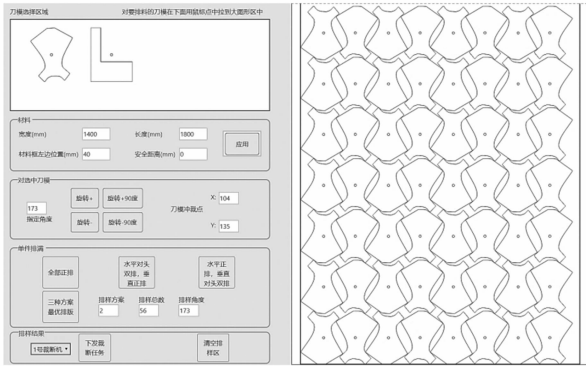


图 9 排样结果显示

当确定最优排样方案后, 选择待运行的裁断机编号并点击下载裁断任务按钮, 系统会将排样区域中第 i 行、 j 列样片旋转中心相对冲裁料原点的二维坐标值 x_{ij} 、 y_{ij} 以及自身旋转角 θ_{ij} 存入后端数据库中, 并发送裁断机编号至本地服务器, 数据库中第一行冲裁样片的排样数据如表 1 所示。

表 1 第一行冲裁样片的排样数据

ID	X	Y	θ
1	113.96	133.09	173
2	269.32	134.23	353
3	435.13	133.09	173
4	590.49	134.23	353
5	756.30	133.09	173
6	911.66	134.23	353
7	1077.47	133.09	173
8	1232.83	134.23	353

本地服务器收到裁断命令后, 会读取云端数据库中的运行数据, 并下发至裁断机控制器, 控制器界面显示了裁断机的运行状态, 并支持用户对裁断机做一些简单的手动操作, 控制器界面如图 10 所示。

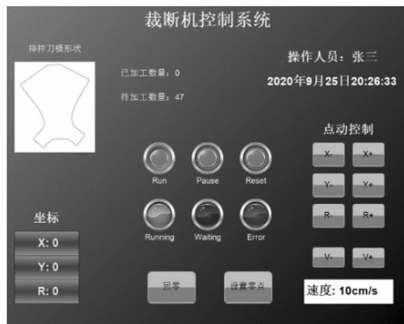


图 10 裁断机控制界面

综上, 测试结果表明本文所设计的系统能够很好的对裁断机进行监控。由表 1 可知, 系统所设计的排样算法能

够生成裁断机所需的运行数据, 并且依靠相应的传输协议可以将数据发送至设备控制器, 实现对冲裁料的加工。

5 结束语

本文设计了一种基于云平台的裁断机监控系统, 在本地服务器基于 Winform 框架搭建了上位机系统, 通过 OPC UA 协议实现了与裁断机控制器的数据交互。在阿里云服务器基于 VUE 框架设计了前端用户操作界面, 并通过 Canvas 标签搭建了排样结果可视化区域。基于 Django 框架设计了后端, 主要编写了刀模排样算法用于生成单一刀模样片在从材料上的排样方案。同时, 通过搭建 MQTT 数据交互模块实现了裁断机控制命令的下发和运行数据的回传。通过该系统, 用户可以完成对裁断机设备的监控。本文下一步研究工作的重点是将裁断机设备的运行数据进行采集并根据数据对设备进行故障分析。

参考文献:

- [1] 王 兴, 侯礼宁, 郭伟洁. 物联网环境下共享公寓系统的研究 [J]. 高技术通讯, 2020, 30 (8): 814-821.
- [2] 许继影, 陈仕军, 郑 晴. 基于两阶段排放算法的矩形件排样优化方法 [J]. 计算机时代, 2020 (5): 13-15.
- [3] Satoshi Fujita, Takeshi Hada. Two-dimensional on-line bin packing problem with rotatable items [J]. Theoretical Computer Science, 2002, 289 (2): 939-952.
- [4] 孙顺苗. 基于数控机床运行数据的状态监测平台研究与开发 [D]. 重庆: 重庆大学, 2019.
- [5] 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.
- [6] 胡晓轩, 朱 琦, 王 浩, 等. 面向船舶制造车间的数据采集与监控系统 [J]. 造船技术, 2020 (4): 68-74.
- [7] 方嘉铭, 于海霞. 基于单片机的机床控制系统 [J]. 电子世界, 2020 (17): 199-200.
- [8] 刘 洋, 刘明哲, 徐皓冬, 等. 基于消息代理的 OPC UA 发布/订阅模式研究与实现 [J]. 高技术通讯, 2018, 28 (6): 553-559.
- [9] 曾周煜东, 何 青. 基于 CodeSys 的车辆组态系统界面的设计 [J]. 自动化技术与应用, 2020, 39 (9): 119-123.
- [10] 龚永罡, 付俊英, 汪昕宇, 等. MQTT 协议在物联网中的应用研究 [J]. 电脑与电信, 2017 (11): 89-91.
- [11] 韩改宁, 李永锋, 高伊腾. 基于嵌入式 Qt 下的 MySQL 数据库设计与开发 [J]. 微型电脑应用, 2020, 36 (5): 25-27.
- [12] 董 朋, 韩佳颖, 杨士龙. 基于 Django 的船舶运动预测系统的设计与实现 [J]. 内燃机与配件, 2020 (15): 206-207.
- [13] 程德昊, 何元清, 蔡春昊. 基于阿里云物联网平台的数据可视化 [J]. 电脑知识与技术, 2020, 16 (22): 50-51.
- [14] 张建中. 基于 Web3D 技术的煤矿 BIM 模型集成及可视化研究 [J]. 煤炭工程, 2020, 52 (7): 161-168.
- [15] 姚振扬, 李聚才, 张 亚, 等. 基于 DXF 文件的压电自适应微细电火花二维 CAM 软件研究 [J]. 制造业自动化, 2020, 42 (2): 46-50.