

# 电器试验设备远程监控系统设计与实现

成建宏, 马训鸣

(西安工程大学 机电工程学院, 西安 710600)

**摘要:** 针对现有电器试验设备监控系统的一些缺陷, 设计了一种基于 Web 的远程监控系统, 用于实时监控设备运行状况和试验数据; 介绍了系统的总体结构和系统功能; 根据系统的功能需求设计了数据库, 实现了系统对不同类型、不同地点的试验设备的通用化; 利用 Redis 缓存降低了磁盘 I/O, 提高了系统的响应速度; 在保证安全的情况下, 实现了对试验设备的远程启停控制; 通过 WebSocket 技术和 Redis 订阅发布模式, 实现了由设备端到客户端的高效数据推送; 利用 ECharts 框架实现了静态和动态试验结果数据的可视化; 经测试, 系统运行效果符合预期, 证明了以上方法的可行性。

**关键词:** 电器试验设备; 远程监控; Redis; WebSocket

## Design and Implementation of Apparatus Testing Device Remote Supervisory System

Cheng Jianhong, Ma Xunming

(College of Mechanical and Electrical Engineering, Xi'an Polytechnic University, Xi'an 710600, China)

**Abstract:** Regarding to some flaws of current supervisory system of apparatus testing device, a kind of remote supervisory system based on Web is designed to monitor the running status of device and test data in real time. General structure and function of system are introduced. According to the function requirement of the system, database is designed to realize the universality of system for testing device of different type and different location. Disk I/O is reduced by Redis cache, and system response speed is increased. Remote start-stop control is realized in the condition of guaranteed security. Realizing the efficient data push from device to client by WebSocket technique and publish-subscribe mode of Redis. Realizing the visualization of static and dynamic testing result data by ECharts framework. After testing, the operation effect of system conforms to expectation which proved the feasibility of above methods.

**Keywords:** apparatus test device; remote supervisory; Redis; WebSocket

## 0 引言

传统的电器试验设备监控系统只基于设备端, 只能在试验现场的设备端获取试验相关信息和对设备进行管理, 这样信息的分散导致信息获取效率和管理效率低下, 因此需要开发集中式监控系统对电器试验设备进行信息集成和集中管理<sup>[1]</sup>。目前, 集中式数据采集与监视系统已广泛应用于电力系统、给水系统、石油、化工等领域<sup>[2]</sup>。然而在低压电器试验场所应用的还比较少, 且存在兼容性差、难扩展、未实现网络化、监控对象受地点限制、客户端受平台限制、客户端维护升级困难等问题<sup>[3-5]</sup>。

本文针对上述问题, 开发了一种基于 web 的电器试验设备远程监控系统, 实现了电器试验设备的信息集成和集中管理, 系统能够实时显示设备状态和试验数据、对设备进行远程启停控制。系统不受时间、地点、平台的限制, 用户可随时随地通过有网络的计算机或移动终端掌握现场的试验状况。针对 B/S 模式实时性差的问题, 采用多种方法有效提高了系统的实时性。在设计中充分考虑了系统的

通用性和可扩展性, 能够兼容常规的电器试验设备, 对于设计过程中未考虑到的特殊试验设备, 只需扩充数据库模型, 即可兼容。经测试, 系统运行良好, 能够满足试验人员和管理人员实时监控设备运转情况和对试验设备进行集中管理的需求。

## 1 系统结构和系统功能

### 1.1 系统结构

系统采用 B/S 和 C/S 相结合的模式, 试验场地局域网内采用 C/S 模式, 这符合局域网内传输数据量大、专用性强的特点; 在 Internet 上采用 B/S 模式, 无需考虑客户端的兼容, 便于对软件进行维护升级。系统总体结构如图 1 所示, 系统由设备端、服务器端和客户端三部分组成。

设备端是系统的数据来源, 电器试验设备中包含工控机、主控板卡、试验电路等部分。主控板卡用于数据采集、试验过程控制。工控机上运行的监控软件通过与主控板卡进行串口通信, 完成对设备运行状况的监视、对试验过程的控制和对试验结果数据的采集, 试验参数和试品信息也在设备端录入, 设备端监控软件将获取到的数据转为固定格式的字符串后通过 Socket 通信发送给服务器端的数据交互程序, 此外还能接收和执行服务器端数据交互程序发送

收稿日期: 2018-07-09; 修回日期: 2018-09-05。

作者简介: 成建宏(1993-), 男, 山西吕梁人, 硕士研究生, 主要从事监控系统、远程自动化方向的研究。

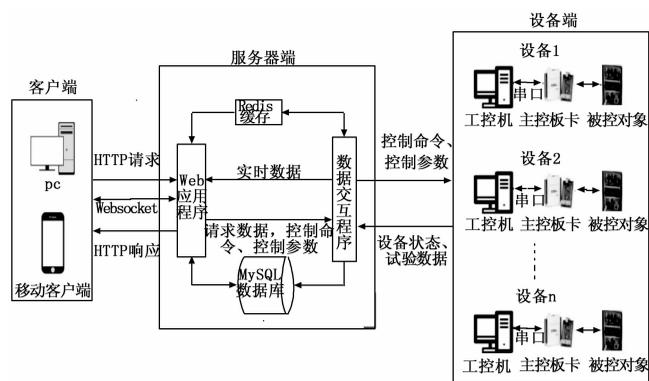


图 1 系统结构

的远程控制命令和试验参数。

服务器端包括数据交互程序、Redis 缓存、MySQL 数据库、Web 应用程序。数据交互程序利用异步 I/O 模块 asyncio 编写, 通过 Socket 与多台试验设备通信, 是 Socket 服务器端, 用于接收设备端发送的数据, 将其中的设备状态数据、试验结果数据解析为 Key-Value 格式的数据后存入 Redis 缓存。如果设备状态数据中包含故障和报警信息, 还需将其存入 MySQL 数据库, 为了减轻数据库压力, 对于正常的状态信息不提供历史状态查询, 不存入数据库。设备端录入的试验参数、试品信息也存入 MySQL 数据库, 同时试验结果数据需定时保存到 MySQL 数据库中, 防止内存被占满。此外, 数据交互程序与 Web 应用程序之间利用 Redis 的订阅发布功能进行通信, 用于接收 Web 应用程序发送的请求数据、控制命令、控制参数, 向 Web 应用程序推送实时数据。Redis 订阅发布模式本质上是共享内存, 这种进程间通信方式避免了对进程中数据的拷贝和传递, 既节约了内存空间又速度快<sup>[6]</sup>。Redis 缓存用于存储频繁更新的实时数据, MySQL 数据库用于存储不经常变动的数据和历史数据。Web 应用程序负责接收客户端发来的 HTTP 请求和返回 HTTP 响应, 并与 Redis 和 MySQL 进行交互, 从 Redis 缓存中获取实时数据, 从 MySQL 数据库中获取不经常变动的“冷数据”。

客户端即计算机或者移动终端中的浏览器, 客户端负责与用户进行交互, 向 Web 服务器发送 HTTP 请求, 接收 Web 服务器返回的响应并将其中的 HTML 文档解析后显示在浏览器上, 将信息展现给用户。若请求的页面中包含实时数据, 还需与 Web 应用程序维持一条 WebSocket 连接, 用于实时数据的推送。

### 1.2 系统功能

系统具有用户管理、设备管理、试验管理、设备监控、报表管理、日志管理等功能, 在此仅对设备监控功能做详细介绍。系统需实现传统工业控制系统中的上位监控功能, 能够实时显示设备运行状态和现场数据, 其中设备运行状态包括设备启停状态、报警信息、故障信息、安全连锁情况、各工位状态信息、控制模式等。现场数据包括各试验场地和实验室信息、各设备基本信息、各工位基本信息、

试验参数、工位上的试品信息、试验结果数据等, 其中试验结果数据采用动态曲线图的形式显示。此外, 系统还能够满足安全连锁条件的情况下对设备进行远程启停控制, 远程控制和设备端控制互锁, 以防冲突。在进行远程控制前需将设备控制模式切换为远程控制模式, 然后输入控制参数, 即可远程启动试验。

## 2 系统设计

系统采用基于 Python 语言的 Django 框架进行开发, 使用的 IDE 为 PyCharm, 编程语言版本为 python3.6.3, Web 框架版本为 Django2.0。使用 MySQL 和 Redis 数据库管理数据, 前端采用 Bootstrap 框架, 图表的绘制采用 Echarts, 这些开发环境均为开源免费软件。下面对系统设计中的部分关键问题分别进行详细分析。

### 2.1 数据库设计

在本系统中, 数据库是一个非常核心的部分, 设备采集和录入的数据需要进行存储, 用户、试验项目、试验场地、试验室等相关信息都要通过后台管理录入数据库, 数据存储架构的合理与否直接影响系统的功能和性能<sup>[7]</sup>。系统采用关系型数据库 MySQL 和 Redis 缓存相结合方式对数据进行管理, MySQL 用于存储不经常变更的“冷数据”, Redis 用于存储读写操作比较频繁的“热数据”, 以满足对数据进行高并发读写的需求。这种数据管理方式既能提高系统响应速度, 又能保证数据的安全。

#### 2.1.1 MySQL 数据库

根据试验设备的实际情况和本系统的功能需求, 在充分考虑系统通用性的前提下, 将数据库分为基础数据库、试验数据库、知识库和系统库 4 个部分。其整体结构如图 2 所示。



图 2 数据库结构

基础数据库主要用于对试验场地、试验室、设备、工位、试验主电路基本信息进行描述。以设备信息表为例, 其基本信息包括所在地、所在实验室、设备编号、生产厂家、设备类型等。由于不同类型的设备, 其工位信息模型一般也不同, 因此需要对不同类型的设备分别建立相应的

工位信息表。

试验数据库用于对试验相关信息进行描述,包括试验项目信息、试品信息、试验参数、试验结果、控制命令、故障记录、报警记录。试验项目表与试品信息表通过外键相关联,在进行试验前需新建试验项目。试验参数表和试验结果表应按照不同的试验类型分别建立。故障记录表中只记录故障设备、故障代码和故障时间,根据故障代码查询知识库获取相应的故障信息,以节省数据库空间。

知识库用于对系统知识进行存储,其中设备库用于记录系统中已经进行建模的各设备类型相关的知识,包括相应的工位信息表表名、可选试验类型等信息。在设备信息表中将设备类型作为外键与设备库相关联,当添加新设备时,只需勾选相应的设备类型,即可获得其相关信息,无需重复输入。比如在设备库中存储工位信息表表名,这样在为设备添加工位时,可根据不同的设备类型,自动关联相应的工位信息表,动态生成不同的 form 表单。元件库用于记录试验主电路中各开关元件的信息,在试验主电路信息表中,将元件作为外键,在建立试验主电路模型时,只需勾选相应元件即可。试验库用于记录在系统中已进行建模的各类型试验相关的知识,包括其对应的试验参数表表名、试验结果表表名、可选设备类型等信息。考虑到有的综合型试验设备可进行多种类型的试验,这意味着同一种试验也可以在多种类型的设备上进行,因此通过在试验库中将可选设备类型设为“多对多”字段,将其与设备库相关联。在试品信息表中,将试验类型作为外键与试验库相关联,在添加试品时,通过勾选试验库中的试验类型添加其试验类型。

系统库用于存储系统相关的信息,包括用户信息表、用户权限表和系统日志。

### 2.1.2 Redis 缓存

根据实时数据的类型,在 Redis 中分别设计了不同的存储结构,如表 1 所示。采用 Hash 类型存储设备状态和工位状态信息,Value 部分为一个 HashMap 对象,该 HashMap 对象的 Key 为设备 ID、工位编号、状态名等拼接成的字符串,Value 为字符串类型的状态数据。而试验结果数据由于添加频繁且需要使用容器盛放,如果采用 Hash 类型存储,每添加一个新数据都需要取出容器对象、反序列化、添加新数据、序列化、更新,开销较大。因此采用 List 类型存储试验结果数据,每一个工位的一个被测物理量对应一个键值对,Key 为“设备 ID-工位编号-变量名”,Value 部分为 List 对象,新的数据直接插入即可。为了防止内存被占满,需定时将旧的试验结果数据存储到 MySQL 数据库,并从 Redis 中删除。

### 2.2 远程控制与安全联锁

系统具有远程启停控制功能,通过点击启动按钮并输入控制参数即可远程启动设备。首先用户需要进入想要控制的设备的监控页面,点击按钮将其控制模式切换为远程控制模式,然后在联锁条件满足的条件下点击启动或停止

表 1 Redis 存储结构

	Key	Value
设备状态	deviceCondition	设备 ID-状态名:Data
工位状态	positionCondition	设备 ID-工位编号-状态名:Data
试验结果数据	设备 ID-工位编号-变量名	[Data <sub>1</sub> ,Data <sub>2</sub> ,...,Data <sub>n</sub> ]
	...	...

按钮,Web 应用程序会生成一条固定格式的控制命令字符串。如果是启动命令,页面会跳转到该工位中的试品对应的试验参数输入界面,用户输入试验参数后点击提交,生成一条固定格式的试验参数字符串,生成的命令和参数经数据交互程序发送给相应设备,设备端监控程序将控制命令和控制参数解析后传给其控制子程序执行,从而实现远程控制。

试验应在满足安全条件的情况下进行,因此远程控制功能也需进行安全联锁。一方面,设备端控制与远程控制需进行互锁,设备的控制模式为设备端控制时,监控界面不显示启停控制按钮;另一方面,需要对试验条件进行前后端验证。在前端验证中,若不满足试验条件,则将启动按钮设为禁用状态;后端验证中若不满足试验条件,则点击启动按钮后重定向到当前页面,并显示警告信息。

### 2.3 动态数据的实时更新

本文采用 WebSocket 技术和 Redis 发布订阅模式实现动态数据的实时显示,WebSocket 是 HTML5 提出的一项浏览器和服务器之间的通信协议,其本质上是 TCP 连接,浏览器和服务器可以利用该协议实现双向通信,使 B/S 模式具备了 C/S 模式的实时通信能力。该方法与传统的 Ajax 轮询技术相比,避免了不必要的请求,传输的数据量少,因此其资源消耗少、实时性强<sup>[8]</sup>。

动态数据实时更新的具体流程为:客户进入包含动态数据的页面时,客户端通过 javascript 代码与服务器建立 WebSocket 连接,并向服务器发送设备 ID、工位编号等请求数据,服务器端的 Web 应用程序接收请求数据后通过 Redis 发布订阅将其发送给数据交互程序,并等待交互程序的数据,交互程序接收到请求数据后,对后续接收的设备端数据进行判断,如果与请求数据相关,则在将其保存到 Redis 缓存的同时,通过 Redis 发布订阅发送给 Web 应用程序,Web 应用程序接收到数据后通过 WebSocket 发送给客户端浏览器,浏览器接收到数据后,对监控页面中的数据进行局部刷新,实现动态数据的实时更新。这种更新方式实现了由设备端主动向客户端推送数据,设备状态未发生变化或没有新的试验数据产生就不会有数据传输,最大程度节省了系统资源和带宽。

### 2.4 数据可视化

电器试验设备监控系统通常需要利用波形图的形式对试验结果数据进行可视化显示,通过对目前常用的几款前端可视化插件进行对比,本文选用 ECharts 实现数据的可视

化, ECharts 是一款基于 HTML5 的图形库, 图形创建简单、支持的图形丰富<sup>[9]</sup>。其具体实现步骤如下:

1) 引入 JS 文件:

```
<script src="/static/js/echarts.js" ></script>
```

2) 准备一个放置图表的容器:

```
<div id=" realtime _ data" style=" width: 800px; height: 500px;" ></div>
```

3) 利用 Ajax 请求或 WebSocket 从后台获取数据:

对于静态试验结果数据, 只需在页面加载完成后, 进行一次 Ajax 请求获取试验数据; 对于动态试验结果数据, 需与 Web 应用程序建立 WebSocket 连接, 获取服务器端推送的新数据。静态数据与动态数据的判断标准为试验是否已完成。

4) 初始化图表, 设置参数, 生成图表:

```
var realtime _ data _ chart = echarts.init ( document.getElementById (" realtime _ data")) //初始化图表
option = { --- }; //设置 option 属性, 将上一步获取的后台数据设置在 option 中
```

```
realtime _ data _ chart.setOption (option, true) //生成图表
```

设置 option 属性和生成图表需放在上一步返回数据之后的回调函数中, 以实现动态刷新效果。

### 3 系统测试

基于以上系统设计, 对系统的设备监控功能模块进行了测试。测试阶段采用个人笔记本电脑上的 Django 框架自带的轻量级 Web 服务器, 其性能能够满足测试环境的需求。所监控的对象是一台脱扣器吸合释放电压测试设备, 其设备端监控系统界面和各工位如图 3 和图 4 所示。

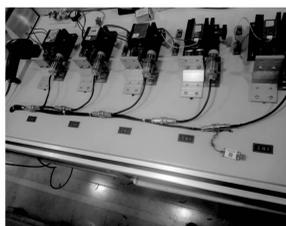


图 3 设备端监控系统

图 4 设备端工位

监控主界面如图 5 所示, 按所在地和试验室对设备进行检索后, 页面显示相应的设备列表, 列表中包含设备类型、启停状态, 控制模式、联锁条件等信息。点击运行状况进入设备主界面, 如图 6 所示。在实时波形列表点击查看, 即可查看设备实时采集到的试验数据, 如图 7 所示, 其横坐标由预设的试验次数决定, 每进行一次试验, 试验曲线中动态增加一组数据。点击试样列中的试样编号可查看试样的相关信息, 包括基本信息、试验参数和试验结果, 如图 8 所示。点击切换为远程控制, 最右侧会新增一列, 用于对设备的远程启停控制。点击开启后在跳转的新页面中输入并提交控制参数, 设备即按照设定的参数运行, 同时

开启按钮变为停止按钮。

测试结果表明, 系统监控模块的各项功能运行状况良好, 可实时地对试验设备进行监控。系统动态数据更新过程无页面刷新和卡顿, 更新响应时间小于 0.5 s, 表明本文所设计的远程监控系统框架能够在满足基本监控功能的同时, 保证了较高的实时性, 基本满足设计目标。



图 5 设备监控主页面

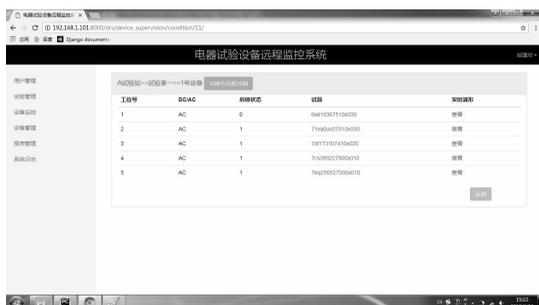


图 6 设备运行状况页面

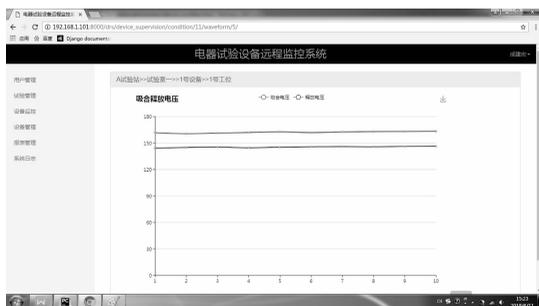


图 7 实时试验数据



图 8 试样信息

## 4 总结

### 4.1 总结

为了解决低压电器试验设备监控系统对信息集成和网

络化集中管理的实际需求,设计了低压电器试验设备远程监控系统。该系统能够实时显示试验现场数据,准确反映现场设备的运行状态,能够对设备进行远程启停控制,具有良好的人机交互界面。在满足以上基本功能的基础上,该系统还具有以下优势:

1) 通用性和可扩展性好:本系统从设计之初就考虑到其通用、可扩展的需求,对于不同类型的设备和试验,分别建立了相应的数据库模型。因此系统对于开发阶段考虑在内的不同类型、不同地点的设备均适用,在开发阶段未考虑到的特殊设备,可以通过扩展数据库的方式使其兼容,因此系统具有较高的可扩展性。

2) 易维护和管理:管理员可通过 Django 自带的后台管理模块对系统进行管理,无需开发额外的后台管理软件,这也是 Django 的优势之一<sup>[10]</sup>,新设备的添加可以通过后台管理轻松完成。系统的升级维护也只需在服务器端进行,对用户的正常使用没有影响,减少了系统维护人员的工作量。

3) 易用性:该系统实现了网络化且不受平台限制,用户可以通过计算机或移动终端中的浏览器随时随地访问系统,无需安装专用的客户端软件,并利用 Redis 缓存、WebSocket 技术和 Redis 共享内存通信等方法减轻了因此带来的实时性变差的问题。

4) 低成本:系统所采用的开发环境均为免费软件,能够大幅度降低系统的开发费用。此外,由于维护方便,系统在以后使用中的维护费用也比较低。

## 4.2 展望

本系统的成功测试给电器试验设备通用远程监控系统的建设提供了一个良好的开端。远程监控所涉及的内容广泛,由于时间和能力有限,很难对其中的每个细节做深入的研究,因此系统还存在诸多不足。在以后的工作中,需要在以下几个方面进行完善和充实。

数据库模型需要进一步完善和扩充。电器试验设备种类繁多,不同类型的试验设备,其工位信息模型不同,不同类型的试验,其试验参数和试验结果模型一般也不相同;同一类型的试验设备由于生产厂家设计理念或应用场景的不同,也会存在差异。这可能导致试验设备与数据库模型不匹配的情况出现,因此需要在今后的运用过程中对数据

库模型不断进行扩充,以提高系统通用性。

建立和完善基于 ActiveX 控件的元件库。目前该监控系统尚未在元件库中添加 ActiveX 控件,因此无法在网页中动态显示试验主电路的状态和试验过程。在以后的工作中需要建立和扩充基于 ActiveX 控件的元件库,在监控页面中嵌入 ActiveX 控件来显示主电路的单线模拟线路图,以实现其试验过程的可视化。这也是一项长期的工作,需要在以后的实际应用中逐步扩充元件库。

添加手动控制功能。电器试验设备上位监控系统的控制功能通常都包含手动控制,因此在远程监控系统中也不应缺失此功能。开发远程手动控制功能之前,应先保证元件库中有相应的元件,并且能够动态模拟试验主电路的状态。

## 参考文献:

- [1] 丁 律, 颜文旭. 远程监控技术及其应用发展 [J]. 化工自动化及仪表, 2016, 43 (6): 563-568.
- [2] 魏智勇, 张 勇. 低压电器试验监控系统设计与通用监控平台开发 [J]. 低压电器, 2009 (5): 54-56, 59.
- [3] 丁志东, 滕成龙, 张 钰, 等. 低压电器大容量试验站网络化监控系统的设计与实现 [J]. 电器与能效管理技术, 2016 (12): 51-55.
- [4] 熊永前, 肖 伟, 宁佐清. 大型开关电器试验站计算机监控系统的设计与实现 [J]. 低压电器, 2006 (10): 43-47.
- [5] 李 旭, 梁伟文, 苗本健. 基于 CC-Link 的低压电器试验监控系统的设计与实现 [J]. 低压电器, 2009 (17): 32-35.
- [6] 任嘉栋. 基于 Redis 的统一监控平台的设计与实现 [D]. 中国科学院大学 (工程管理与信息技术学院), 2016.
- [7] 赵 炯, 盛 凡, 周 杰, 等. 设备故障监测与诊断系统数据库设计 [J]. 机械设计与制造, 2014 (5): 155-158.
- [8] 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.
- [9] Li D Q, Mei H H, Shen Y, et al. ECharts: A declarative framework for rapid construction of web-based visualization [J]. Visual Informatics, 2018.
- [10] 龚新定, 余艳梅, 吴小强, 等. 基于 Django 的实验室信息管理系统设计 [J]. 微型机与应用, 2016, 35 (22): 108-111.
- [11] 高林奎, 樊戈平, 王德钊, 等. 激光伺服跟踪轨距测量装置 [J]. 中国铁道科学, 1994, 15 (2): 79-95.
- [12] Bladon P, Bowling K, Braren H, et al. The challenges of integrating novel wayside rolling stock monitoring technologies, a case study [Z]. IHHA, Perth, 2015.
- [13] Tsai H C. Railway track inspection based on the vibration response to a scheduled train and the Hilbert-Huang transform [J]. Proceedings of the Institution of Mechanical Engineers 2015, 229 (7): 815-829.
- [14] 植立才. 基于 CP IV 轨道三维检测系统的新型轨检小车结构设计与分析 [D]. 广东: 广东工业大学, 2016.
- [15] 蒲广益. ANSYS WORKBENCH 基础教程与实例 [M]. 北京: 中国水利水电出版社, 2010.

(上接第 46 页)

- [7] 李或雯. 轨道部件图像检测系统设计 [D]. 南昌: 南昌大学, 2012.
- [8] 陈利娟. 非接触轨距检测系统设计 [D]. 成都: 西南交通大学, 2008.
- [9] 刘 军, 陈翠梅, 卢峰华. 高速综合检测列车轨道检测梁制造技术研究 [J]. 高速动车组技术, 2013 (s1): 50-53.
- [10] 刘 伟. 轨道巡检悬挂系统结构设计以及强度分析 [J]. 上海铁道科技, 2015 (4): 29-30.