
杭州电子科技大学

硕士学位论文

题目：变电站在线监测系统设计及关键技术研究

研究生 杜坤

专业 集成电路工程

指导教师 陈科明 副教授

完成日期 2021 年 3 月

杭州电子科技大学硕士学位论文

变电站在线监测系统设计及关键技术研究

研 究 生： 杜坤

指导教师： 陈科明 副教授

2021 年 3 月

**Dissertation Submitted to Hangzhou Dianzi University
for the Degree of Master**

**Design and key technology research of
substation online monitoring system**

Candidate: Du Kun

Supervisor: Associate Prof. Chen Keming

March, 2021

杭州电子科技大学

学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：

日期：2021 年 3 月 12 日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署单位名称仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：

日期：2021 年 3 月 12 日

指导教师签名：

日期：2021 年 3 月 12 日

摘要

电网规模的持续扩大，对变电站的安全稳定运行提出了挑战。目前变电站的在线监测方式已取代传统人工巡检方式，但仍存在一定的不足：当前设备故障检测只针对设备实时运行状况进行检测，无法预测故障发生时间或预测精度不足；组态软件开发维护成本高、跨平台兼容性差，客户端界面难以及时同步被控设备的运行数据状态。

本文针对目前变电站监测技术上存在的以上缺点，基于物联网技术设计了一套变电站在线监测系统：将自主开发的智能网关对接变电站 PLC 控制柜，实时采集设备生产数据；通过 4G 将数据上传至云平台进行分析处理；设计 Web 实现设备运行状态、故障告警、人员管理等信息的可视化。通过在线监测系统实现了变电站的远程监测、配电图 Web 组态等需求。

为了能够准确预测设备故障发生时间，针对目前基于卡尔曼滤波算法的故障预测方案初始噪声矩阵参数人为设定不准确、滤波估计过程中不能自适应调整等缺点导致的预估精度不足问题，进行改进研究，提出了一种基于双策略灰狼优化算法(Multi-Strategy Gray Wolf Optimizer, MGWO)卡尔曼滤波算法的故障预测方案。首先引入非线性收敛因子、记忆位置更新方程双策略，来提高传统灰狼优化算法(Gray Wolf Optimizer, GWO)全局勘探和局部开采的平衡能力，通过改进后的 MGWO 算法对卡尔曼滤波噪声矩阵参数进行动态自适应调整，降低卡尔曼滤波算法的发散性，进而提高预估精度。测试表明，预估精度的均方根误差相比传统卡尔曼滤波算法减小了 60%，提高了设备故障预测准确度。

针对目前组态软件开发成本高、跨平台兼容性差、服务器与浏览器之间通信延时较长等问题，提出了一种基于 SVG 与 WebSocket 的 Web 组态设计方法。利用 SVG 设计组态编辑模块的图形元素，并通过有向图理论简化组合图形多输入多输出的复杂逻辑运算，降低了复杂图形的开发成本；利用 WebSocket 通信技术使得设备状态数据能够实时推送至 Web 组态界面。测试表明，可有效降低开发成本，提高兼容性，并能达到服务器向浏览器实时推送信息数据的目的。

关键词：变电站监测系统、故障预测、卡尔曼滤波、Web 组态

ABSTRACT

With the continuous expansion of power grid scale, the safe and stable operation of substation is challenged. At present, the on-line monitoring method of substation has replaced the traditional manual inspection method, but there are still some shortcomings: the current equipment fault detection only detects the real-time operation status of the equipment, and can not predict the time of failure or the prediction accuracy is insufficient; the development and maintenance cost of configuration software is high, and the cross platform compatibility is poor, and the client interface is difficult to synchronize the operation data of the controlled equipment in time Status.

In view of the above shortcomings in the current substation monitoring technology, this paper designs a set of substation online monitoring system based on the Internet of Things Technology: connect the self-developed intelligent gateway with the PLC control cabinet of the substation to collect the equipment production data in real time; upload the data to the cloud platform for analysis and processing through 4G; design the Web to realize the equipment operation status, fault alarm, personnel management visualization of information. Through the on-line monitoring system, the users can monitor the requirements of system and structure the Web configuration remotely.

In order to accurately predict the occurrence time of equipment failure, aiming at the shortcomings of the current fault prediction scheme based on Kalman filter algorithm, such as inaccurate initial noise matrix parameters and unable to self-adaptive adjustment in the process of filtering and estimation. In this paper, a fault prediction scheme based on Multi-Strategy Gray Wolf Optimization (MGWO) Kalman filter algorithm is proposed to improvement the accuracy of fault prediction. Firstly, propose a nonlinear convergence factor and memory position updating equation to improve the balance ability of the traditional Gray Wolf Optimizer (GWO) in global exploration and partial extracting. The improved MGWO algorithm adjusts the parameters of Kalman filter noise matrix adaptively to reduce the divergence of Kalman filter algorithm and improve the prediction accuracy. The test results show that the Root Mean Square Error of the prediction accuracy is reduced by 60% compared with the traditional Kalman filter algorithm, and the equipment fault prediction accuracy is improved.

Aiming at the problems of high development cost, poor cross platform compatibility and long communication delay between server and browser, a Web configuration design method based on SVG and WebSocket is proposed. SVG is used to design the graphic elements of configuration editing module, and the complex logic operation of multiple-input multiple-output is simplified by directed

graph theory, and the development cost of complex graphics is reduced; the equipment status data can be pushed to the Web configuration interface in real time by using WebSocket communication technology. The results show that it can effectively reduce the development cost, improve the compatibility, and achieve the purpose of real-time push information data from the server to the browser.

Keyword: Substation monitoring system, IOT, Kalman filtering, Web configuration

目 录

摘 要.....	I
ABSTRACT.....	II
目 录.....	IV
第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 变电站在线监测系统研究.....	2
1.2.2 故障预测技术研究.....	2
1.2.3 Web 组态技术研究.....	3
1.3 论文结构安排.....	4
第二章 变电站在线监测系统设计.....	6
2.1 系统总体设计.....	6
2.1.1 需求分析.....	6
2.1.2 系统总体框架.....	6
2.2 智能网关设计.....	8
2.2.1 智能网关硬件设计.....	8
2.2.2 智能网关软件设计.....	9
2.2.3 通信协议设计.....	12
2.2.4 智能网关实物展示.....	13
2.3 云平台设计.....	14
2.3.1 设备云.....	15
2.3.2 业务云.....	15
2.4 Web 端设计.....	17
2.4.1 开发模式分析.....	17
2.4.2 技术架构设计.....	18
2.4.3 Web 功能实现.....	19
2.5 本章小结.....	27

第三章 基于 MGWO 卡尔曼滤波算法的故障预测方案.....	29
3.1 整体设计思路	30
3.2 卡尔曼滤波原理及误差分析	31
3.3 GWO 算法及其改进策略	32
3.3.1 基本 GMO	32
3.3.2 非线性收敛因子策略	34
3.3.3 记忆位置更新方程策略	34
3.3.4 GWO 算法优化测试.....	35
3.4 基于 MGWO 的卡尔曼滤波算法优化设计	36
3.4.1 适应度函数选取	37
3.4.2 算法优化流程	37
3.5 基于 MGWO 卡尔曼滤波算法的故障预测	38
3.6 实验测试	40
3.7 本章小结	43
第四章 基于 SVG 与 WebSocket 的 Web 组态设计方法.....	44
4.1 基于 SVG 的 Web 组态编辑模块设计	45
4.1.1 Web 组态编辑模块总体设计	45
4.1.2 基于 SVG 的图元编辑设计	47
4.1.3 基于有向图的组合模型算法设计及实现	49
4.2 基于 WebSocket 的 Web 组态数据通信设计	53
4.2.1 WebSocket 概述	54
4.2.2 WebSocket 数据通信设计	55
4.2.3 WebSocket 浏览器端实现	57
4.3 实验测试	57
4.3.1 功能测试	57
4.3.2 兼容性测试	58
4.4 本章小结	59
第五章 总结与展望.....	60
5.1 总结	60
5.2 展望	60

致 谢.....	62
参考文献.....	63
附录.....	66
作者在读期间发表的学术论文及参加的科研项目	66

第一章 绪论

1.1 研究背景及意义

据国家统计局数据，2010 年至 2019 年全国供电量逐年增长，统计数据如图，2018 年、2019 年均超过 70000 亿千瓦时^[1]。

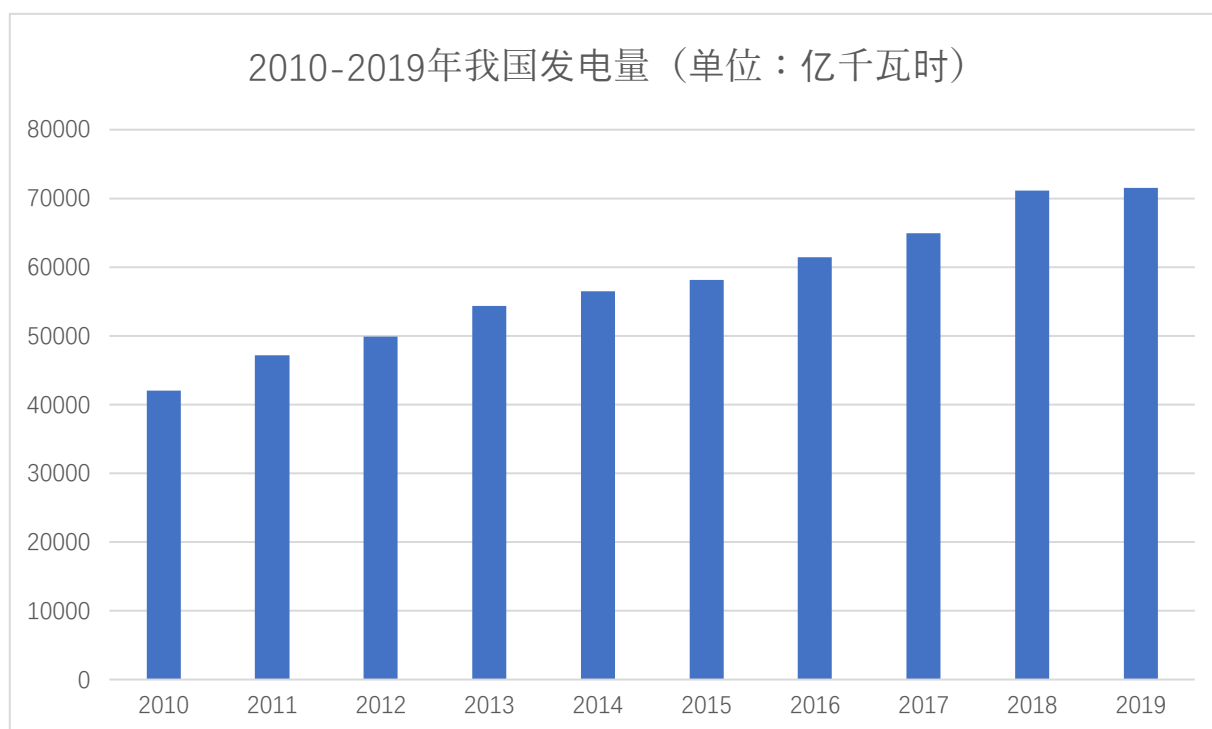


图 1.1 2010-2019 年我国发电量

全国电力生产、电网规模的高速发展，运用物联网、大数据等新一代信息技术对传统人工巡检变电站系统的方式进行智能化升级已经成为一种必然的发展趋势。”十二五”、”十三五”规划中明确提出要推进智能电网建设，而变电站系统运行情况监测、故障预测、配电网可视化等是其中重要的技术创新单元，所以开展关于变电站在线监测系统的研究具有重要意义^[2]。

目前已有不少将变电站系统监测与物联网技术相结合的系统落地，但这些系统仍存在设备异常状态监测难、数据通信延时高、软件服务不完善等问题。因此，设计一种能够实时监测变电站系统运行状况、功能完善、设备故障可预测、用户交互体验好、易于维护的变电站在线监测系统显得尤为重要。

本文设计实现了一套变电站在线监测系统，以满足用户对变电站系统的远程监测、数据分析需求，同时对系统实现的两个关键方面进行深入研究优化：(1)变电站系统的正常运行对设备故障检测提出了准确性、及时性、可预估的要求，需要设计故障预测算法，不仅要能够

实时监测还应能对设备运行异常状况进行预测；(2)为便于用户可视化监测变电站状态，需要设计界面相应快、强扩展性、安全稳定、具有良好跨平台兼容性、界面与云端数据同步实时性高的 Web 组态软件来支持用户构建配电图。

1.2 国内外研究现状

1.2.1 变电站在线监测系统研究

随着“智能电网”概念的普及，国内外涌现了不少变电站监测的智能化系统^[3]。在国内，张强基于组件化设计思想设计了一种变电站监测管理系统，降低了系统软件的开发成本，但没有设计设备故障监测算法，电力设备安全隐患没有降低^[4]；王丽娟利用 GPRS 技术、SOA 架构模式实现了电力设备数据入网，并对用户用电行为信息进行分析，但没有有效提高设备状态与客户端画面同步的及时性^[5]；高阳采用 B/S 架构模式开发了一套基于 Web 的电力设备远程监控系统，但没有针对变电站配电图设计组态界面，系统定制化、通用性程度低，而且只能实现设备故障的实时监测，无法预估故障未来发生时间^[6]。国外也已逐步对电力系统的各个方面进行升级改造：德国将新型载波抄表系统创新性的应用到远距离传输技术中；美国则针对用电管理系统的规模扩大进行研究；韩国则从电网系统的软件服务满意度着手^[7]。

根据研究和实际市场调研，归纳出当前变电站监测系统在以下方面存在不足：

(1) 上层应用软件设计落后、业务功能涵盖不全。没有针对配电图组态界面、系统关键参数信息大屏可视化等界面进行设计，用户不能直观清晰地监测设备运行状态；未考虑到用户对于随时随地查看变电站系统运行状况需求，只提供了 PC 端应用软件。

(2) 电力设备、线路故障监测机制不完善甚至没有，只是对运行数据进行远程展示，变电站监测系统的安全运行的核心诉求是设备故障实时监测与预警。

(3) 数据通信传输技术落后，不能将现场运行数据、设备异常状况及时同步到远程客户端展示。

1.2.2 故障预测技术研究

故障预测，是在设备运行时持续检查并判断设备运行情况，并通过传感数据采集、算法运算处理分析来预测设备运行状态^[8]。通过传感器采集到的变电站系统的变压器温度、变压器油位、电压、电流以及变电站环境因素等数据信息，来反映设备运行状况。

变电站运行系统的故障预测方法大致可以归纳为以下三种^[9]：

(1) 基于数据分析

此类方法主要根据经验、系统模型，建立输入与观测输出的映射关系，通过实测值的分析来映射观测值，继而据观测值预测未来状态。通过对已有采样数据进行训练分析，求解出未知样本，继而实现故障预测。

(2) 基于概率统计

不同于基于数据分析的预测方法，而是采用建立概率密度函数来建立概率模型，求出未来某个时刻处于某种状态的概率信息，以此来进行故障预测。通过基于马尔可夫链的预测方法是比较常用的概率预测算法，通过建立系统某一时刻与更早时刻的马尔可夫性，进而以当前状态观测结果来预估未来时刻状态^[10]。

(3) 基于系统模型的故障原理

该类方法主要通过对故障机理建立故障预测模型，然后针对实际电气系统状态进行评估预测。卡尔曼滤波算法由于其算法结构清晰且性能优异，被广泛应用于故障预测、通信、控制、制导、定位等领域。但其算法性能强依赖于算法初始参数值，通常采用人工根据经验设定的方式。近年来出现了不少对卡尔曼算法的改进研究，文献[11]通过引入重复采样和粒子权重的方式，提高了卡尔曼算法的预估精度；傅建平基于无迹粒子滤波理论来预测锂离子电池剩余寿命^[12]。

总之，电网规模的扩大加快了变电站系统故障预测理论的成熟，不同预测算法的适应场景不同，需要选择合适算法并加以优化改进，最终实现故障预测的目的，并结合实时检测算法，确保系统运行异常状况的实时监测与预测。

1.2.3 Web 组态技术研究

组态一词源于“Configuration”，是指用户利用计算机，根据现场设备连接信息，通过简单的拖拽构建出软件界面，以达到直观监控设备的目的，故也被称为“二次开发平台”^[13]。组态的概念从提出到现在，已有多类组态软件产生，表 1.1 简单概括了目前国内外主流软件情况。

表 1.1 国内外主流态软件情况

软件名称	厂家	优点	缺点
WinCC	德国 Siemens	环境完善、功能完备	可操作性要求高
iFix	美国 Intellution	汉化、集成度高	额外功能付费
Citech	澳大利亚 CIT	性能强	兼容性低、操作难
InTouch	美国 Wonderware	图元库丰富、市场占有率高	费用高
力控	三维力控	支持数据批量存储、分布式数据库	兼容性低、实时性低
Controx	华富远科	分布式报警、图表种类齐全	稳定性、系统响应性低
组态王	亚控	集成度高、告警自存储	兼容性低

近年来，Web 技术发展迅猛，传统组态软件的开发维护难、监控效率低下使得人们开始尝试将组态技术和 Web 技术有机结合，出现了“Web 组态”的概念^[14]。Web 组态技术摒弃传统组态软件的 C/S 架构，采用 B/S 架构，降低了开发成本，使得前后端开发人员工作职能解耦，

并支持远程随时随地组态，并且由于基于浏览器端，其跨平台兼容性得到了有效保障。

Web 组态技术主要包括两大模块：组态图形界面、服务器与 Web 组态实时数据通信。如何开发出体量小、扩展性高、稳定性强、兼容性高的图形界面，以及如何提高组态界面与设备数据同步的及时性是 Web 组态技术的研究重点。目前网页图形绘制技术和服务器如表 1.2 所示。

表 1.2 网页图形绘制技术

技术类型	特点
Flash	针对广告、多媒体设计而推出，适合动画处理，需要安装插件
Silverlight	主要用来设计多媒体、网络交互应用，性能强大，需要安装插件
Java	功能丰富，开发流程复杂，跨平台兼容性低
Canvas	无法支持事件处理，渲染性能差，开发简便
SVG	不依赖分辨率，无失真放大，DOM 可操作性强，适合动态交互，轻量

目前服务器与 Web 组态实时数据通信方式如表 1.3 所示。

表 1.3 服务器-Web 组态实时数据通信方式

通信技术	底层模式	通信特点
Ajax 短轮询	Http 短连接	Web 通过 JavaScript 定时器的方式周期性请求服务器，近似获取实时数据
Ajax 长轮询	线程阻塞型 Http 短连接	维持连接直到服务器返回数据，期间服务器线程被阻塞，不能响应其他请求
SSE	Http 长连接	支持断线重连机制，只支持服务器向 Web 推送数据
WebSocket	基于 Tcp 协议的全双工通信	Html5 新增通信协议，支持浏览器服务器双向通信
Iframe 流	Http 长连接	Web 端内嵌 Iframe 标签并基于 src 属性模拟服务器推送

在进行技术方案选型时，要充分结合自身项目需求、系统开发成本等因素。张洋基于组态王软件开发了上位机组态系统，实现了与 PLC 的有机结合^[15]；程功结合 Html5 和 Ajax 通信技术，采用 B/S 架构模式完成了一套针对煤矿设备的 Web 组态监控系统，降低了软件系统开发维护成本^[16]；田娜利用 SVG、React、Node.js 等技术完成了一套组态监控系统，并借助 RabbitMQ 插件提高了数据接收与推送的实时性^[17]。

1.3 论文结构安排

本文针对传统变电站监测系统的多项不足，通过对智能网关、云平台、Web 应用程序的设计开发，完成了一套智能变电站在线监测系统，为用户提供远程监测、故障检测告警、大屏可视化管理、Web 组态等多项智慧服务。结合智能系统的稳定性、扩展性、实时性、完整性等技术特点，对整个系统进行多层次拆分，并详细分析设计了各个模块，不仅实现了变电站系统的远程在线监测，并着重优化系统性能：提出一种基于 MGWO 的卡尔曼滤波故障预

测方案，实现了设备运行异常状况的实时监测及预测，有效保障了变电站系统的正常运行；基于 SVG 与 WebSocket 实现了配电图 Web 组态，提高了设备数据状态与组态界面同步的实时性，缩短了组态构建周期。

第一章 绪论。首先介绍了变电站监测系统的发展现状及趋势，指出当前系统存在的不足。接着分别从变电站监测系统、故障预测、Web 组态三个方面，阐述了国内外研究现状，确立了本文的研究方向与研究任务。

第二章 变电站在线监测系统设计。首先分析了系统实际需求，介绍了系统整体架构，将系统划分为硬件系统、云平台、客户端三个模块，并由下至上从设备端到客户端进行详细设计阐述，最后展示各个应用实现成果。

第三章 基于 MGWO 卡尔曼滤波算法的故障预测。指出目前变电站监测系统故障预测精确度不够，对卡尔曼滤波算法误差进行分析。提出非线性收敛因子、记忆位置更新方程双策略改进 GWO 算法，提高了 GWO 寻优精度。通过改进后 GWO 算法(记为 MGWO 算法)，不断迭代求出卡尔曼滤波算法的最佳系统初始参数，提高卡尔曼滤波算法预估精度，进而保障了变电站运行系统故障预测的准确度。最后，进行相关测试，测试表明，该故障预测方法能高效准确地预测变电站系统运行异常情况。

第四章 基于 SVG 与 WebSocket 的 Web 组态设计研究及实现。首先分析 Web 组态技术在变电站在线监测系统落地的可行性与现有方式的不足，针对 Web 组态设计两大核心模块进行优化改进：采样 SVG 作为组态软件图元格式，设计实现 Web 组态编辑模块，针对简单图元组合复杂图形出现的多输入多输出复杂的逻辑运算，设计基于有向图理论的组合模型算法，简化了逻辑运算；基于 WebSocket 实现 Web 组态的实时数据推送。最后，对该设计方法进行相关测试，测试表明，本文提出的 Web 组态设计方法开发便捷、跨平台兼容性高、数据通信延时较低。

第五章 对全文的系统设计和关键技术研究加以总结，对系统设计的不足做出分析，提出对未来的改进和发展方向。

第二章 变电站在线监测系统设计

针对当前变电站监测系统故障报警不完善、可视化能力差、智能化程度不足等缺陷，本文设计并实现了一种基于物联网技术的变电站在线监测系统。通过自主研发的智能网关与变电站控制柜 PLC 连接，实时获取传感器采集的变压器温度、油位、电流和环境温湿度等数据并通过 4G 传输方式实时上报到云平台，云平台对数据处理分析之后，将设备运行状态信息、故障等信息发送到客户端应用程序，实现用户对变电站系统的在线监测。

2.1 系统总体设计

2.1.1 需求分析

根据构建变电站监测系统的关键要素以及目前智能变电站监测系统存在的缺陷，对于需要设计的变电站监测系统，需要满足以下几点需求：

- (1) 设备入网，通过智能网关实时上报传感数据至云平台。
- (2) 保证数据通信链路的稳定性，保证设备采集数据和客户端展示数据的一致性与及时性，提高故障检测准确性与报警的及时性。
- (3) 开发配套软件服务，用户可基于此客户端软件实现变电站监测的可视化管理和便捷操作，实现与监测系统智能交互、数据价值的最大化利用。

2.1.2 系统总体框架

根据上述系统需求和设计原则，本文设计了变电站在线监测系统，系统整体架构如图 2.1 所示。

智能网关充当变电站 PLC 控制柜与远程设备云平台的数据桥梁，包括 MCU、通信模块等；远程云平台充当智能网关与云平台的数据桥梁，包括业务处理、数据存储、数据分析处理等内容；客户端直面用户，提供给用户直观的显示和操作功能。以下对这三个部分进行详细介绍：

(1) 智能网关部分。通过 RS232/485 接口、Modbus(RTU)通信协议将智能网关与变电站 PLC 控制柜完成接口协议对接，采集所需现场设备的数据，之后网关通过 4G 传输方式将数据上报到设备云平台，建立设备的远程长连接，保持稳定实时的数据收发。

(2) 云平台部分。分为设备云和业务云两大模块。设备云选用中国移动 OneNET 第三方云平台，智能网关通过 OneNET 提供的通信接口，完成设备上传数据的存储、监控管理以及指令下发；业务云主要负责设备管理、用户管理等具体业务服务，开放应用程序编程接口供客户端调用实现部分客户端功能，并可以与设备云进行通信。需要强调的是，只有在设备与

设备云平台之间才存在与设备之间的数据交互。

(3) 客户端部分。提供给用户最直接的便捷远程操作可视化界面，开发了 Web 网页给用户全面的功能支持，包括配电图组态构建、设备管理、故障预测与报警、用户信息管理、历史与当前数据查询、地图显示等功能。

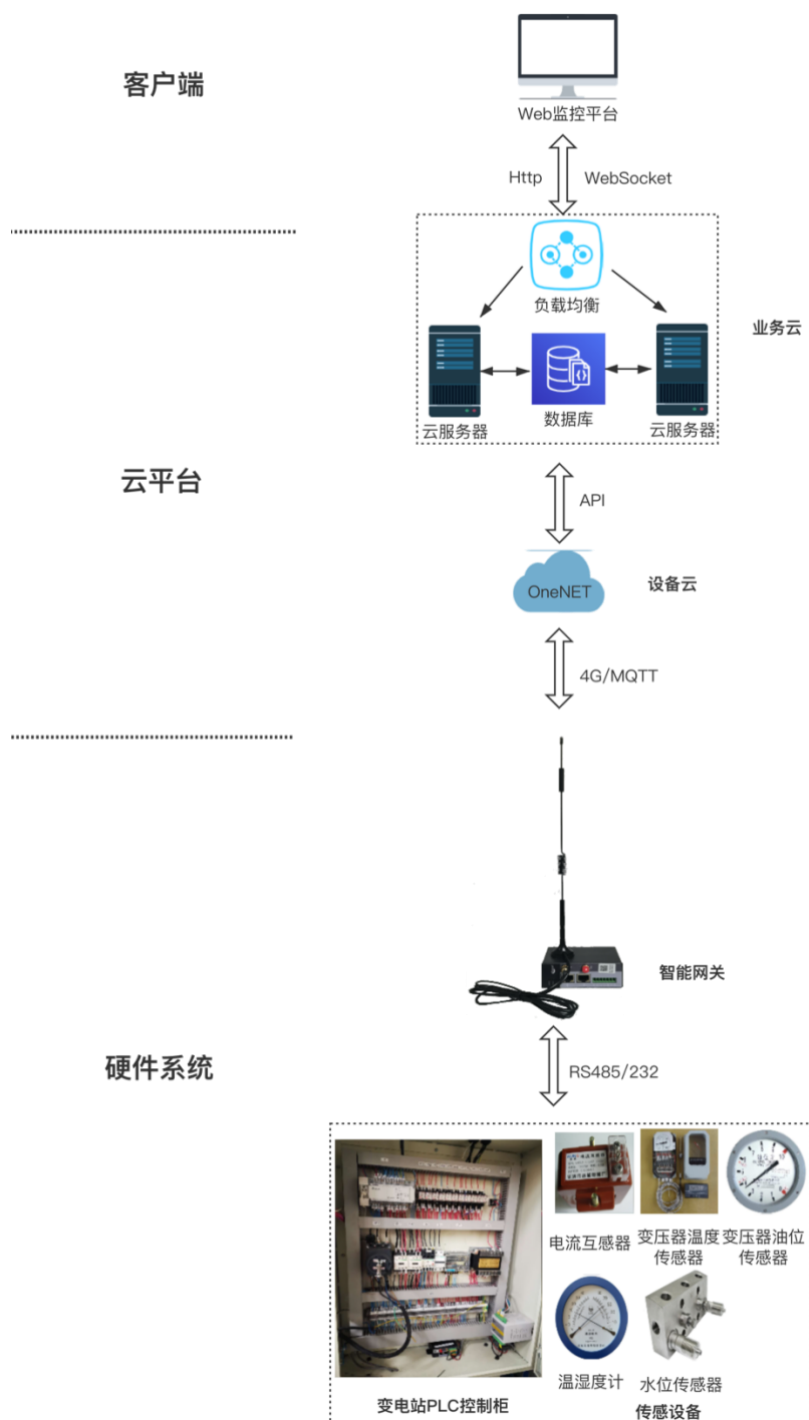


图 2.1 系统整体架构

2.2 智能网关设计

智能网关在本文所设计的配电站监测系统中充当下层设备数据传输的通信枢纽，向上与远程设备云平台建立长连接，向下与本地物理设备连接。本文所设计的网关集成 WIFI、以太网、4G 等多种联网模式，通用性较强，可与远程设备云平台的数据交互指令完成上报、下发操作，这是向上的部分；向下的部分则集成 RS485 模块、RS232 模块、串口隔离模块等多种不同的接口，用来与本地 PLC、传感器等设备的数据交互。

2.2.1 智能网关硬件设计

智能网关硬件架构如图 2.2 所示，网关硬件主要包括 JS9331 模块、MCU 模块、LoRa 射频模块、RS232/485 通信模块、串口隔离模块、4G 通信模块等。

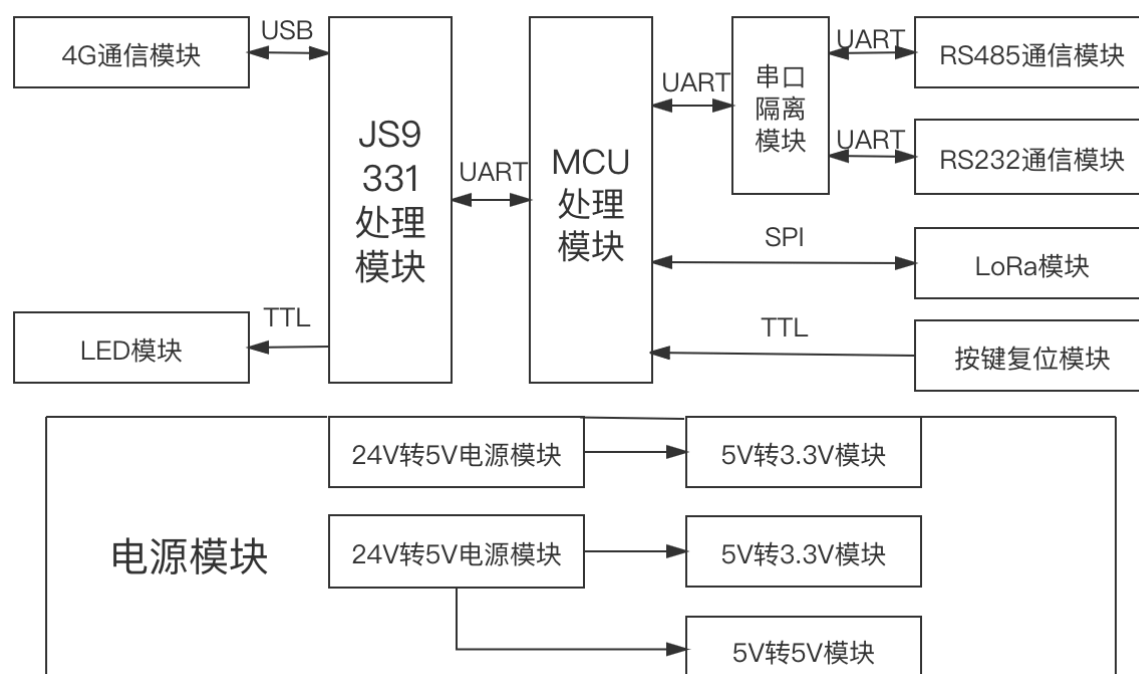


图 2.2 智能网关硬件架构图

MCU 处理模块采用 STM32F103RCT6，汇总 LoRa 射频模块、RS232 通信模块和 RS485 通信模块过来的数据并且转发给 JS9331 处理模块进行处理；JS9331 处理模块一方面汇总处理各个模块的数据信息，另一方面也可以通过 WIFI 或者以太网与服务器平台进行数据通信。

4G 通信模块通过 USB 与 JS9331 处理模块相连，4G 通信模块主要与服务器平台建立通信，以此达到数据的上传和下发。LoRa 射频模块通过 SPI 接口与 MCU 处理模块互连，可以实时收取终端从设备采集到的环境信息，并把数据反馈给 MCU 处理模块，也可以通过 LoRa 射频模块远距离下发指令给终端从设备。

RS232/485 通信模块通过串口与串口隔离模块连接，保证用户主机或者测试系统可以与网关进行数据交互，RS232 电路原理图如图 2.3 所示。

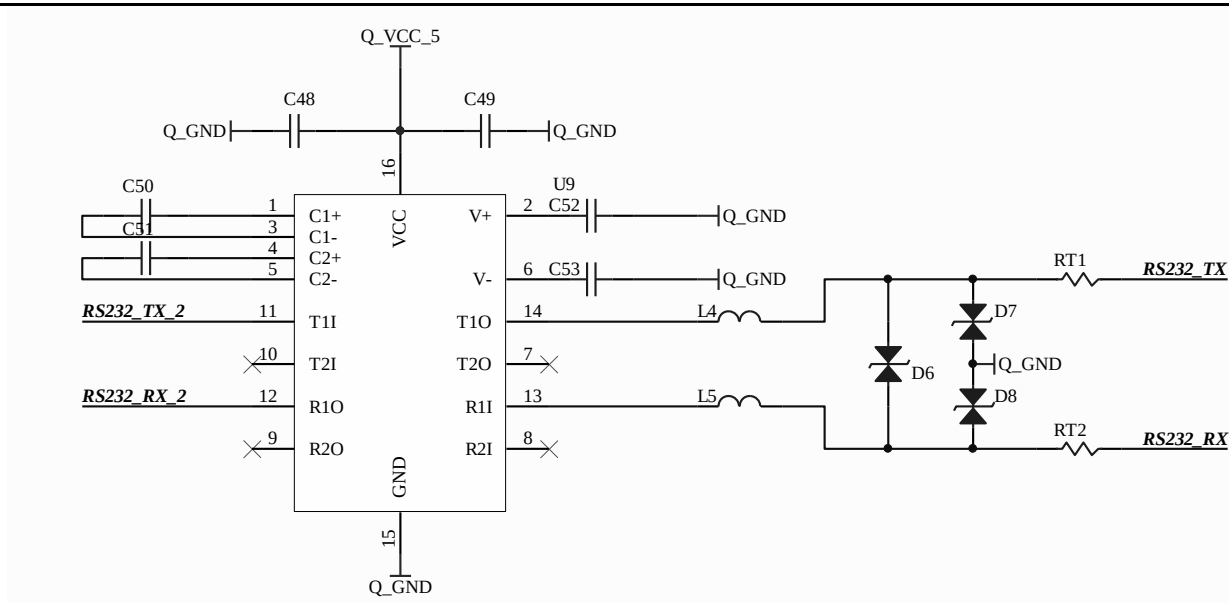


图 2.3 RS232 模块电路原理图

RS485 电路原理图如图 2.4 所示。

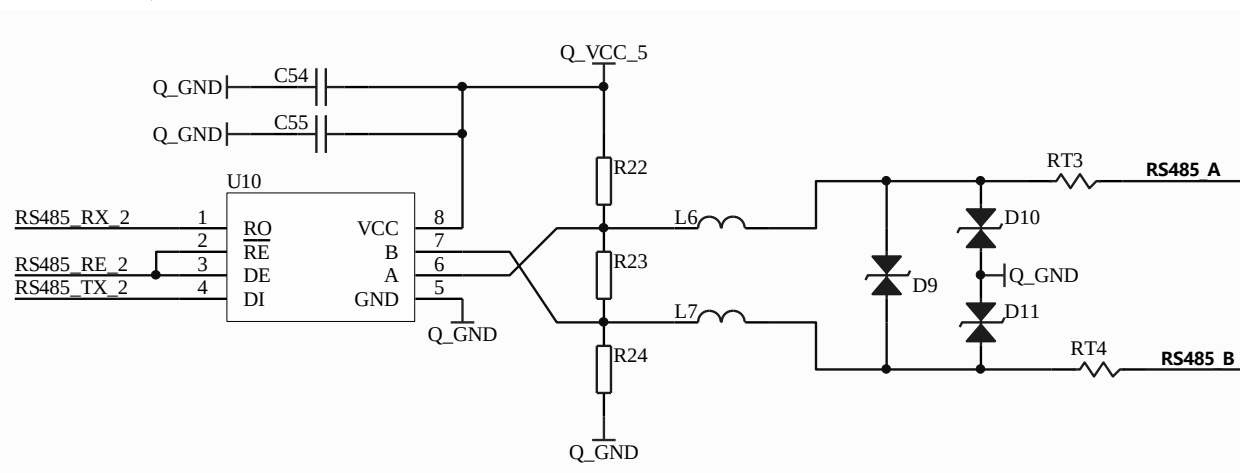


图 2.4 RS485 模块电路原理图

2.2.2 智能网关软件设计

智能网关充当现场设备与设备云平台之间的数据桥梁，是底层设备层面的核心模块，在本文所设计的智能监控系统中，需要满足以下功能：

- (1) 与设备云平台保持长连接并保证数据交互的实时性、稳定性。
- (2) 可定时查询监控点数据并上报到设备云平台。
- (3) 汇聚所有监控节点数据，并可通过下发指令对各个节点进行控制。

智能网关的整体工作流程如图 2.5 所示。

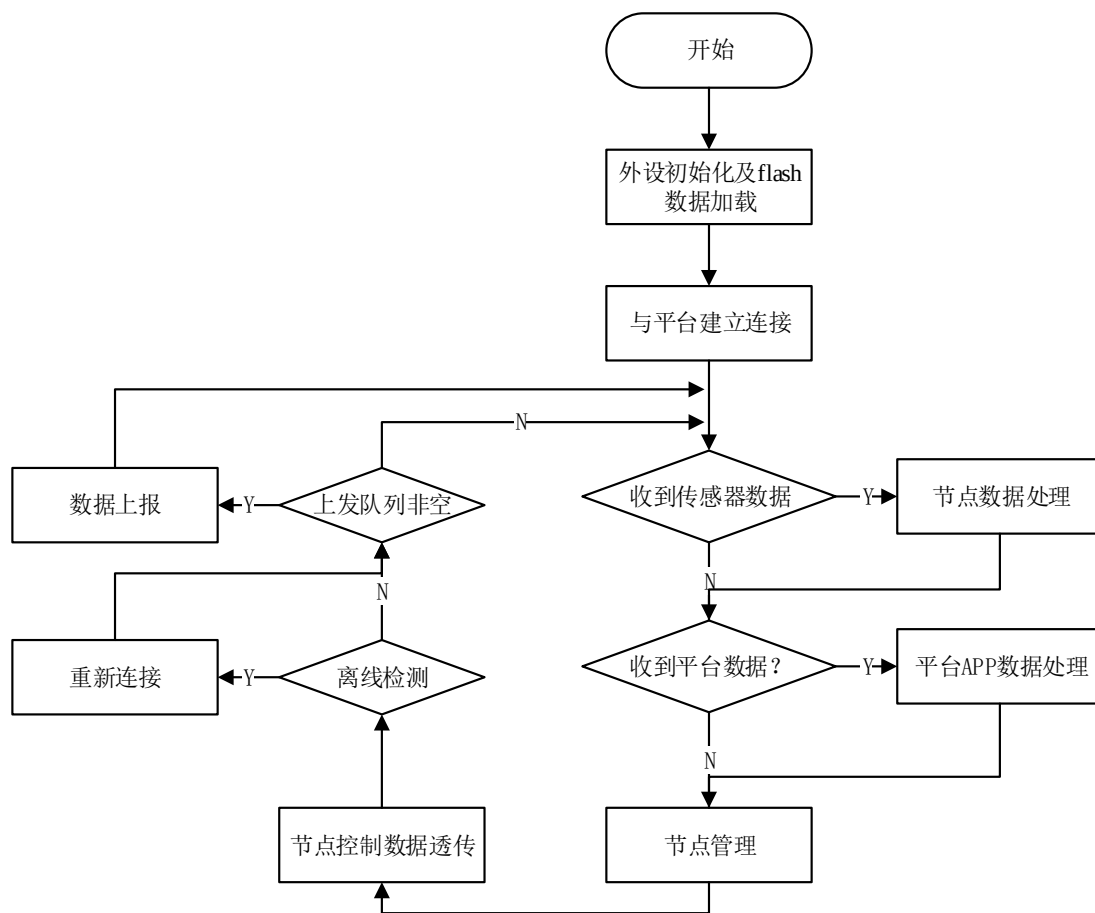


图 2.5 智能网关工作流程图

在该系统设备端与设备云平台两者的交互设计中，主要维护以下几个关键名词：

- (1) 数据流上发队列：暂存底层传感器采集的数据，并排队等待被发送到设备云平台。
- (2) 监控点列表：每个待监控数据的关键信息存储单元集合，设计有掉电保存功能。智能网关通过轮询机制，可定时向查询对应数据流的监控点信息。通过 Modbus(DTU)协议，设备云平台可下发指令来增删改查监控点。
- (3) 监控点数据缓冲区：网关接收到传感数据后立即将其缓存。
- (4) 数据重发队列：在设备到智能网关、智能网关到设备云平台的通信链路中均设有重发机制，来保证数据传输的稳定性。通过 CRC 校验确认是否重发，维护一个队列来暂存待重发数据信息，当下一个时隙间隔到来时重新发送。

OpenWrt 是嵌入式 LINUX 发行版，具有开源和应用广泛的特点。本系统所设计的网关，就是基于该操作系统进行编写，该操作系统常被用作路由器的开发，因此具有以太网和 WIFI 的联网功能，用户也可以自己裁剪，安装和编写相应的程序，实现对应的功能。在操作系统中进行了 4G 模块的挂载，实现了 4G 拨号上网的功能。自主编写了应用程序、网络选择、监测应用程序工作状态和错误收集等脚本。

应用层程序主要由以下部分构成，如下图 2.6 所示。主要由 4 个线程实现，OneNET 的

数据收发线程、串口的数据收发线程、485 的数据收发线程和信息处理的线程组成，实现了图中的功能。

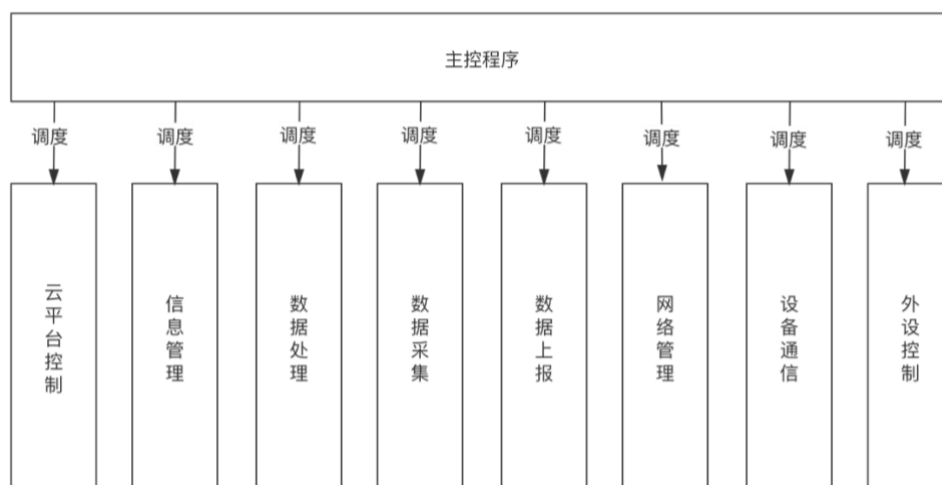


图 2.6 网关软件总体框图

网络选择处理的脚本，以及相应异常处理流程。判断网络是否连接成功，使用 ping 指令，去 ping 指定的域名服务器，中国移动、中国联通和中国电信通用的 DNS 是 114.114.114.114，也是本次设计中 ping 的 DNS。当需要对网络进行重启时，需要执行/etc/init.d/network restart 指令，进行重启网络的操作。具体如图 2.7 所示。

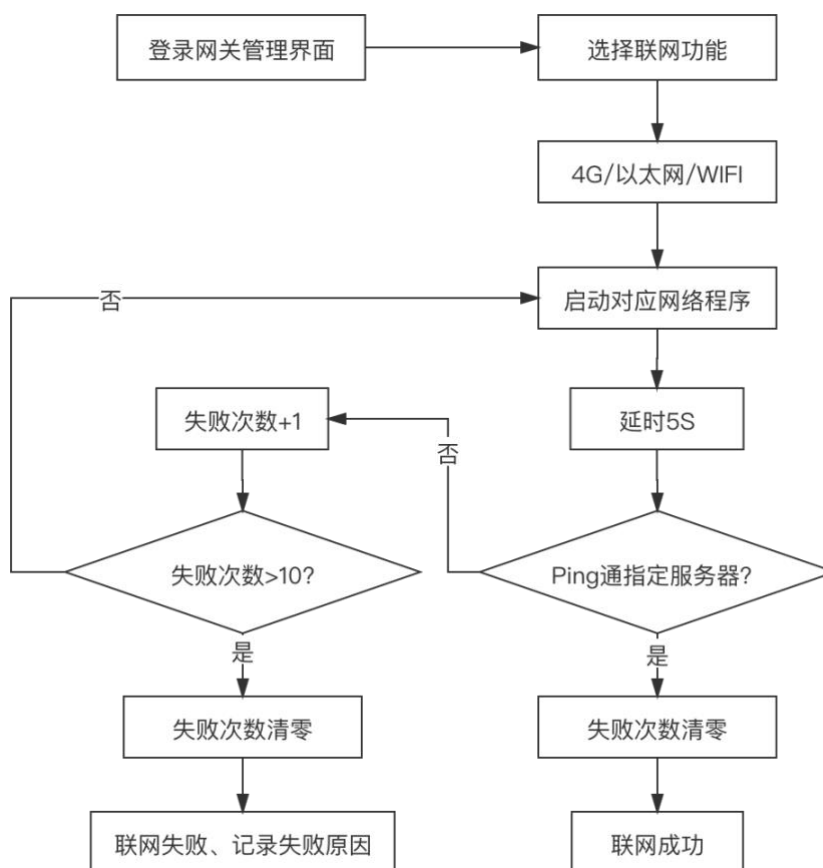


图 2.7 网关联网流程图

错误收集脚本，智能网关在运行过程中，会有相应的日志文件，当程序意外运行崩溃时，会产生 CORE 文件。当应用程序意外终止时，会触发相应的信号，错误脚本开始执行，将日志文件和 CORE 文件打包成压缩包，使用文件传输协议上报给相应的后台。如图 2.8 所示。

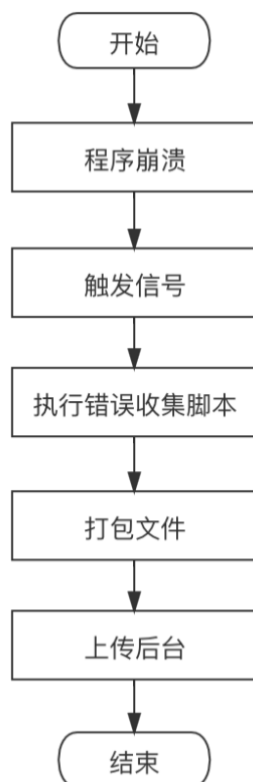


图 2.8 错误收集流程图

2.2.3 通信协议设计

Modbus 技术用于设备和智能终端的通信，对传统工业设备生产数据的采集有着很好的稳定、高效性，并且业界生态圈较好、技术支持比较全面。因此，本文系统选择使用 Modbus 通信协议来实现设备与智能网关、智能网关与设备云平台的通信。

RS485 采用半双工工作方式，在物联网自动化领域应用广泛^[18]。设备与智能网关通过 RS485 总线进行物理连接，并基于 Modbus(RTU)协议传输信息，下表 2.1 展示了帧格式设计。

表 2.1 RTU 传输协议帧格式

设备地址	功能码	数据	CRC16 低 8 位	CRC16 高 8 位
1 字节	1 字节	起始地址，寄存器数 (最大 252 字节)	1 字节	1 字节

本文选用中国移动 OneNET 作为设备云平台，以 OneNET 作为服务主机，采用 OneNET 自提供封装的 Modbus 协议作为物联网设备的通信方式，通信过程如图 2.9 所示：

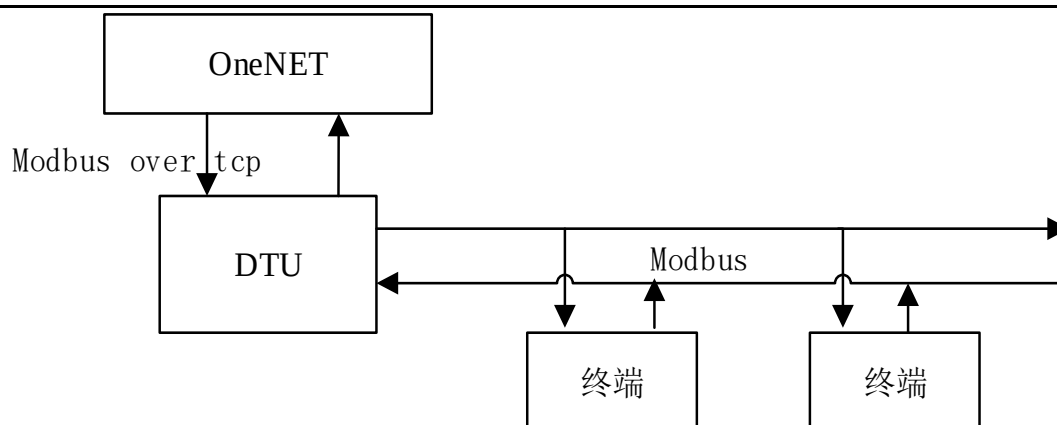


图 2.9 网关和设备云平台通信过程

通过 Modbus(ASCII)协议实现设备云平台与智能网关的通信传输,设计帧格式如表 2.2:

表 2.2 Modbus (ASCII) 帧格式

设备地址	功能码	数据	LRC	结束
1 字符	2 字符	起始地址, 寄存器数 (最大 2*252 字符)	2 字符	2 字符 0X0D 0X0A

总结智能网关与设备云平台的通信设计特点如下:

(1) 针对设备数据的预处理和指令下发, OneNET 云平台提供了两种方式: 一是通过可视化管理界面进行在线便捷操作; 二是暴露出基于 http 协议的数据流处理 API 接口供代码程序直接调用。

(2) OneNET 云平台会根据预先自定义的设备数据流-监控点模型将设备终端上报的数据自动映射。并且提供 API 接口供上层客户端查询数据流的最新数据点, 对应的就是设备的最新数据。

(3) OneNET 云平台支持周期性下发指令来实时查询设备最新数据和状态, 并可以海量存储设备数据。

(4) 智能网关与 OneNET 云平台维持基于 TCP 的 Modbus 协议长连接。

2.2.4 智能网关实物展示

图 2.10 为网关实物图, 图 2.11 为网关电路板。



图 2.10 网关实物图



图 2.11 网关电路板

图 2.12 为网关与变电站 PLC 控制柜连接图，图 2.13 为网关接口图。

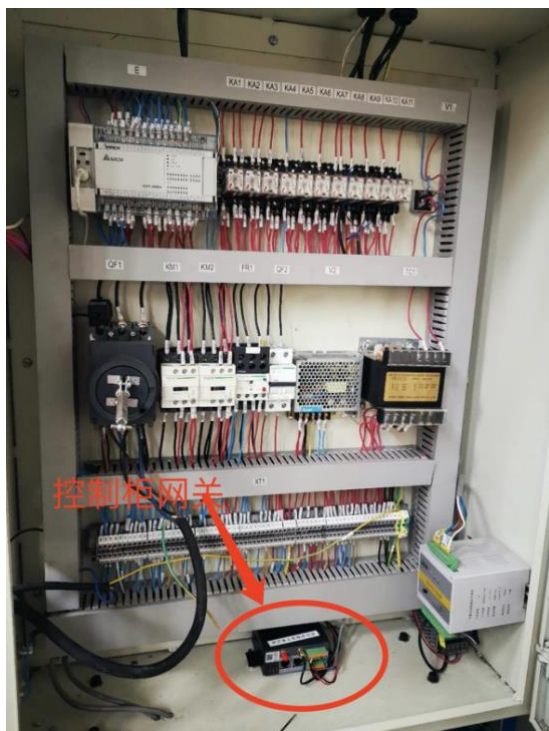


图 2.12 网关与变电站 PLC 控制柜连接图



图 2.13 网关接口图

2.3 云平台设计

客户端与设备之间的通信需要云平台这个媒介，提供设备、用户等信息数据存储以及业务逻辑处理等服务。整个智能监控系统的高性能、高稳定性决定了云平台的设计必须满足高效、安全、稳定的要求。本系统所构建的云平台分为两个部分：设备云平台、业务云平台。设备云负责设备数据收集存储，指令下发；业务云负责逻辑处理、具体业务服务，主要对接的是客户端。而两个云端的交互通过设备云自提供的 API。

2.3.1 设备云

智能网关采集的数据需要传输到远程云端并在云端进行相关数据处理。考虑到设备入网的难度、降低开发成本，经过调研本文决定采用第三方物联网云平台，即中国移动 OneNET 平台。

OneNET 设备云平台对下与智能网关建立远程长连接，并基于 Modbus (DTU) 通信协议进行数据交互、指令上报与下发。其管理界面如图 2.14 所示，供用户直观地进行产品管理、数据流设置、报警触发器设置等实用操作。

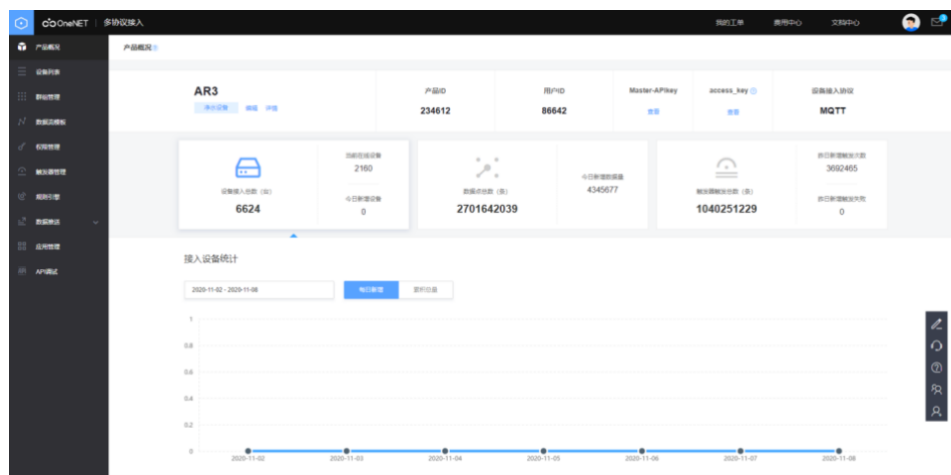


图 2.14 设备云平台管理界面

2.3.2 业务云

2.3.2.1 业务云技术架构

云服务器选用两台阿里云虚拟主机(ECS)来自主构建，负载均衡通过阿里云高可靠弹性负载均衡机制，采用最小连接数轮询负载算法。采用主从存储方式并基于阿里云的 MySQL 云端数据库实现数据存储服务。业务云整体架构如图 2.15 所示。

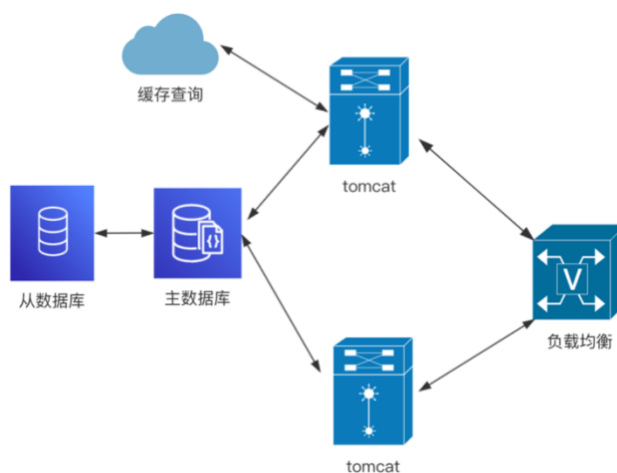


图 2.15 业务云架构

通过自主构建业务云有如下几点优势：

(1) 高灵活性。通过自主设计的业务云平台，可以在使用过程中灵活多变地及时调整业务架构以应对用户需求的变更。

(2) 强功能定制性。本文根据用户需求，结合变电站运行监测必备因素等客观环境，在监控系统平台中开发了报警推送、历史数据管理、设备管理、用户权限管理、配电图等一系列功能。

(3) 速迭代性。为了满足二次开发的快速高效性需求，我们在业务云平台针对常见需求和功能中开放了诸如添加、删除等通用性 API 接口。

(4) 低数据量。业务云平台侧重点与设备云平台不同，主要负责用户具体业务服务，数据量较小，不需要像设备云平台那样需要保持海量设备的长连接以及海量设备数据的存储。

我们在整个开发过程中采用前后端分离的模式，实现前后端的业务、职能解耦，通过 Http 协议进行数据的增删改查。

Spring Boot 有着配置流程简单、易上手的优点，可轻松创建一个生产级别的 Spring 应用，内置 Tomcat、jetty 等 Web 容器^[19]。基于 Spring Boot 框架开发，可以将后端开发人员从繁重无意义的环境配置中解放出来，专注于相关的业务、逻辑算法开发。

本文选用阿里云 Mysql 云数据库，并基于 Mybatis 框架进行数据库的调用，Mybatis 主要维持了数据表中的表结构-业务云服务器中的 DAO 层的一种映射关系，并基于面向对象的思想简化了原本数据库的复杂操作过程，业务云平台的整体架构如图 2.16 所示：

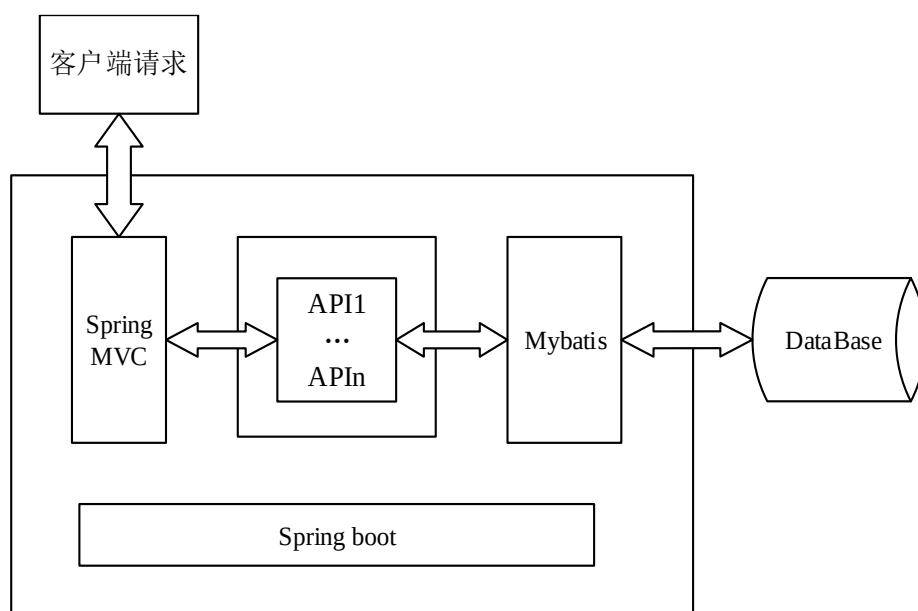


图 2.16 业务云平台的整体架构

2.3.2.2 接口调用鉴权机制

网络请求分为有状态和无状态两种，而基于 http 协议的网络请求是无状态的，因此需要

手动给客户端对服务器的每一次请求加上身份认证信息，这就要求我们需要实现一种接口调用安全认证机制来验证请求来源是否可信。本文在综合考虑市面上几种常用鉴权机制之后，决定通过 JWT 机制来校验 API 无状态条件下的安全性。JWT 是一种基于 JSON 数据格式的令牌(token)判定权限方案，现已被广泛应用于互联网接口通信加密中^[20]。

结合本系统应用场景，设计了一种安全可靠的认证机制，如图 2.17 所示。

(1) 用户登陆成功后，业务云服务器将 http 请求携带的用户唯一标识 `userId` 作为关键参数并通过加密算法生成该用户的专属 token(令牌)，然后返回给服务请求者 `userId` 和 token。

(2) 之后客户端发起的所有网络请求头 header 都要携带 `userId` 和 token，服务器会通过约定好的解密算法对 token 进行解密得到其中的 `userId` 和 token 过期时间，如果未过期，则进一步将传入 `userId` 和解密得到的 `userId` 进行对比，如果相同则认为验证通过，否则需要重新登陆。

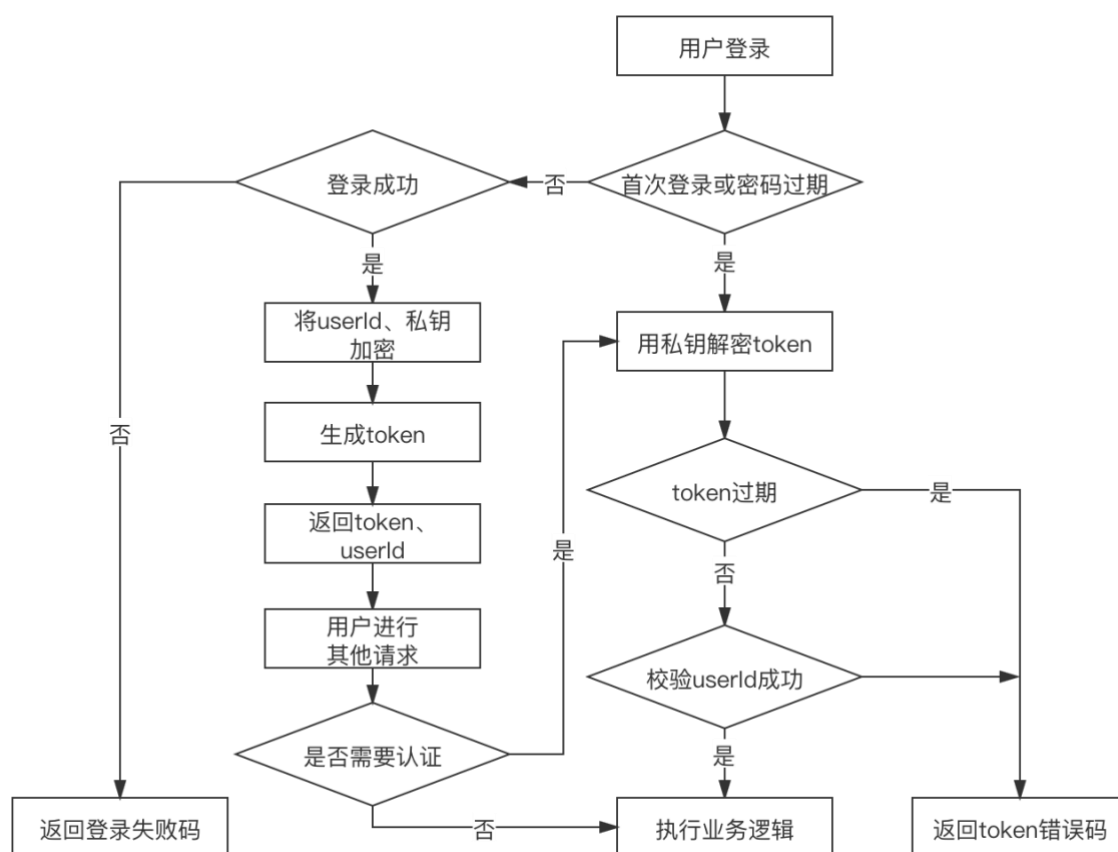


图 2.17 接口调用鉴权流程

2.4 Web 端设计

2.4.1 开发模式分析

Web 端在架构模式选择上有 C/S 模式、B/S 模式两种，B/S 模式和 C/S 模式都是为了使得软件工程内部分工协作、专注于服务使用者而逐渐发展成的一种业界标准工作模式，但 B/S 模式与 C/S 模式的主要不同之处在于后台服务器承载了更多的工作负载，尽可能降低浏览器端的事务压力，而只需专注于与用户进行业务交互，所以 B/S 模式也被成为称为“瘦客户端模式”。对比传统的 C/S 模式有着如下优势：

(1) 低开发维护成本。软件系统逐渐云化，传统的一次性购买软件的模式已经行不通，用户购买的软件服务可以基于浏览器去开发，浏览器只是充当用户服务的“窗口”。

(2) 高迭代性。市场变化、职能演变使得用户需求变更越来越频繁，这对软件产品提出了版本迭代快速性和升级简便性要求。C/S 模式下的软件产品，每一次迭代升级需要厂商同时发布客户端、服务端的更新包；而在 B/S 模式下，只需要服务端单方面发布升级版本即可，不需要用户进行任何操作以及购买软件升级版本包，用户可直接感知到服务端的更新变化。

因此，本文选择 B/S 模式对变电站监测系统进行前后端分离开发。

2.4.2 技术架构设计

客户端直面用户，需要满足使用体验清爽简单、功能实用、系统稳定等条件。针对用户对变电站监测系统的功能需求，本系统软件设计主要集中在设备状态数据监控模块、事件管理模块、配电图模块、电力数据日/月报查询模块、设备管理模块等，除此之外，额外增加了用户管理、设备地图显示、大屏可视化等一系列功能模块来完善系统需求。

根据监测系统 Web 端“高内聚、低耦合”的设计理念^[21]，进一步划分、封装了前端业务部分，分割成一个个独立的小业务模块，各个模块彼此之间数据通信通过 API 接口来实现。

将 Web 前端工程结构拆分为路由模块、组件模块、第三方组件模块、网络请求系统模块、状态管理等，如图 2.18 所示。

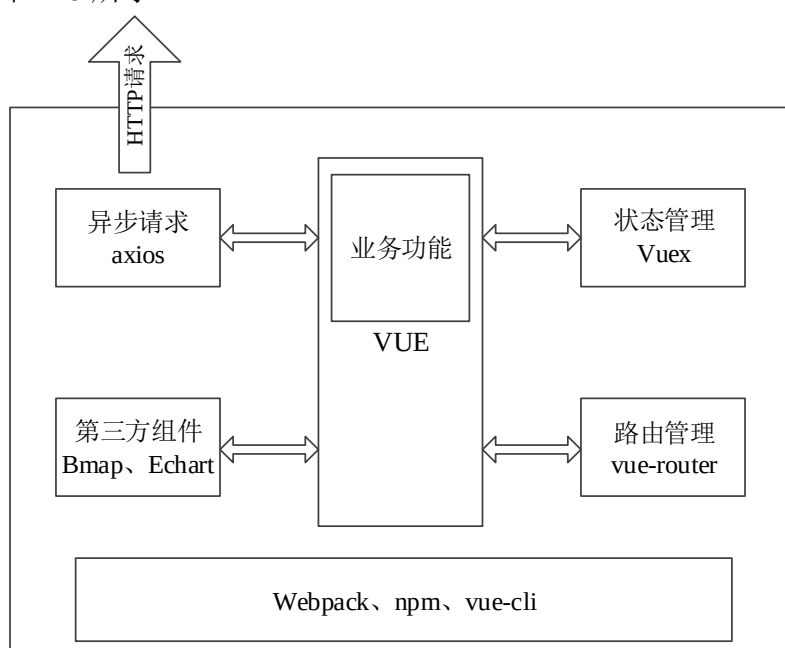


图 2.18 Web 端技术架构

2.4.3 Web 功能实现

2.4.3.1 功能模块结构设计

结合本项目的业务需求，在 Web 网站实现了 Web 组态、用户管理、事件记录、运行环境监测等功能，功能结构如下图 2.19 所示。

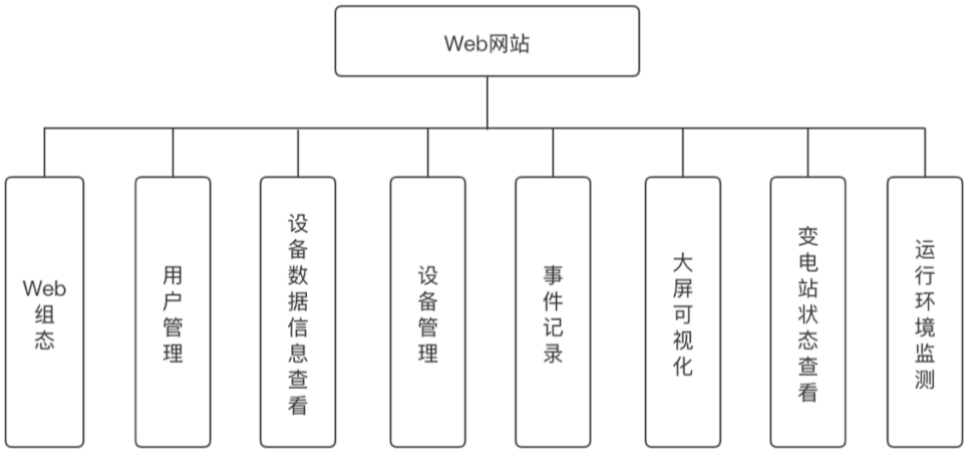


图 2.19 Web 网站功能结构图

2.4.3.2 Web 组态

配电图即用户根据现场设备接线图构建成 Web 组态界面，便于用户直接查看电力设备运行情况，包括三大模块：组态编辑子模块、Web 发布子模块、组态运行子模块。组态架构整体设计如图 2.20 所示：

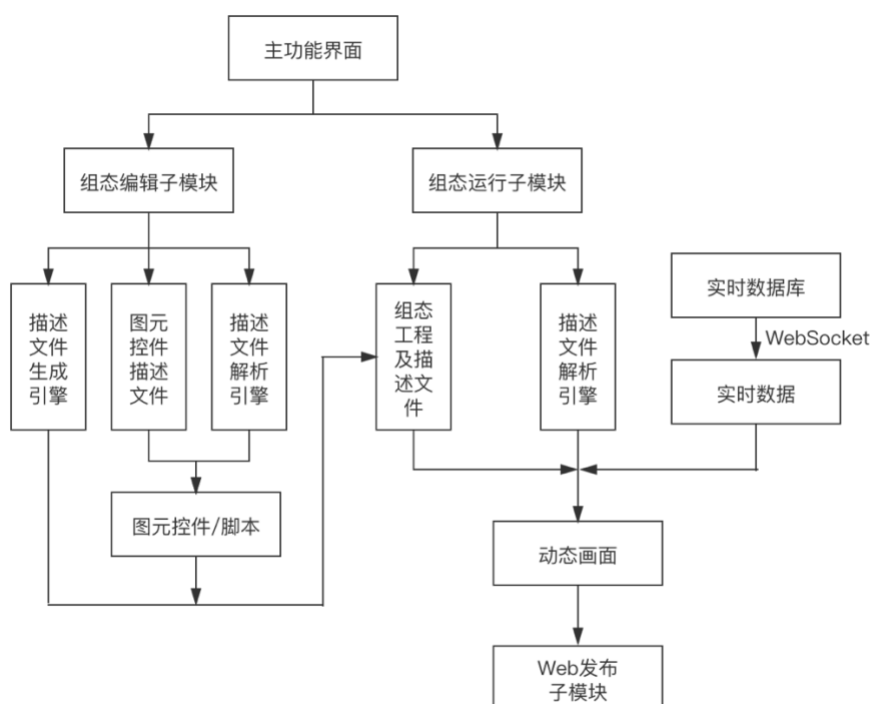


图 2.20 组态软件架构设计

(1) 组态编辑子模块。通过 SVG 文档对象模型的可操作性、可交互性，在前端浏览器中实现组态编辑功能，并设计常用基础图元供用户直接编辑、合成配电图，完成一套跨平台跨终端的可视化组态编辑界面，来实现电气接线图的映射。

(2) 组态运行子模块。服务端数据通过 WebSocket 通信协议实时下发数据给 Web 组态，组态实时展示设备运行状态。

(3) 组态发布子模块。浏览器端采用 B/S 架构完成组态编辑之后，生成组态配置文件，发布到云端，支持远程实时查看编辑。

配电图组态实现效果界面如图 2.21 所示。

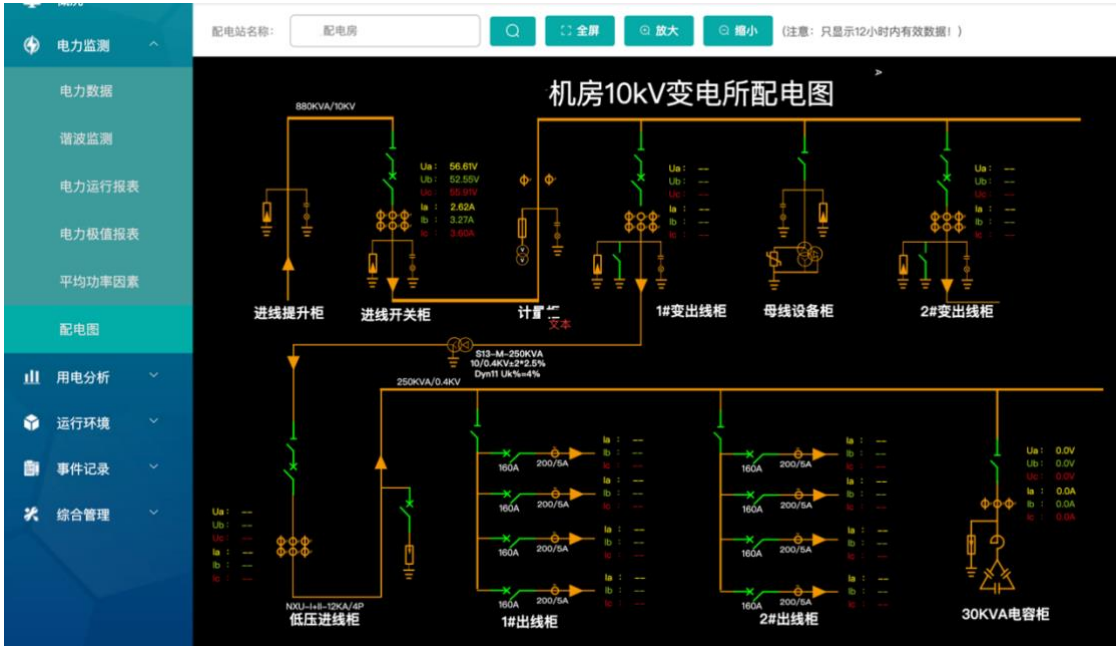


图 2.21 配电图组态界面

2.4.3.3 用户管理

针对用户部分相关业务，功能结构如图 2.22 所示：

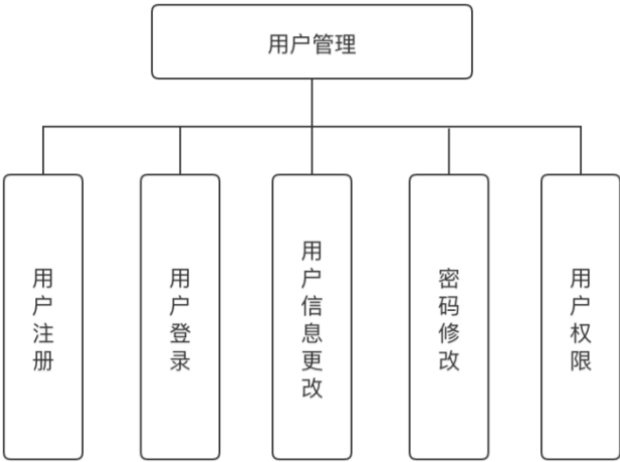


图 2.22 用户管理功能结构

各功能模块详细功能设计如表 2.3 所示。

表 2.3 用户管理功能设计表

条目	设计内容
登录、注册	为每个用户分配身份唯一标识符 <code>userId</code> ；支持邮箱、手机号等方式注册和登录。
用户信息更改	通过校验 <code>userId</code> ，更改姓名、头像、性别等基本信息。
密码更改、忘记密码	根据用户名、第三方通信平台下发验证码验证。
用户权限设置	以设备为维度按照权限级别从高至低依次划分为：拥有者、管理员、工程师

、记录员四种角色。

拥有者是设备最初最绑定的用户；管理员可对设备进行修改、读写操作；工程师对设备可读可写；记录员只能对设备进行读取操作。

实现效果如图 2.23，图 2.24 所示。

图 2.23 用户信息

图 2.24 用户权限设置

2.4.3.4 设备数据信息查看

设备数据信息查看分为实时查看和历史数据查看两部分。其中实时查看功能通过配电图来形象生动地反应设备状态变化及设备实时数据；历史数据支持当日查看和逐日查看，并且两种查看方式均支持列表和曲线两种展示方式，给客户最直观生动的体验。支持设备数据报表导出功能，不仅可以将报表打印出来，便于设备数据留存和查看，而且可以方便未来对接其他第三方数据分析平台。表格和曲线分别如图 2.25、图 2.26 所示。



图 2.25 设备历史数据表格展示



图 2.26 设备历史数据曲线展示

2.4.3.5 设备管理

设备管理模块能够使得用户对设备进行归档管理，包括设备添加、删除、编辑功能。其详细设计如表 2.4 所示。

表 2.4 设备档案功能设计表

条目	内容
设备添加	1.为每台物理设备设计分配唯一身份标识码 S/N 序列号以及对应密码； 2.支持安装时间、设备名称、安装位置等辅助信息编辑； 3.设计四种状态码标识设备状态：01-在线、02-离线、03-连接超时暂时离线、04-服务器未注册设备 Id
设备删除	1.根据用户权限 userRole、设备码 deviceId 校验是否可删除； 2.设计回收站功能，避免用户误删除
设备编辑	支持设备名称、设备位置等基本信息修改，强制不可更改设备唯一身份标识码

图 2.27 展示了基于 Vue 前端框架实现的设备管理 Web 功能界面。



图 2.27 设备管理 Web 功能界面

2.4.3.6 事件记录

事件记录模块的涵盖实时事件、事件记录、网关通信状态三个子功能模块，Web 实现界面如图 2.28 所示。



图 2.28 事件记录 Web 功能界面

实时事件功能模块呈现的是电站环境、配电变压器状态、发动机等设备实时状态和设备异常状况表，记录了近期故障事件的告警类型、位置、发生时间、解除时间、操作人员、故障原因等信息，可支持查看最近半年的事件记录，并可以导出报表。

网关通信状态功能模块记录了安装在各处的智能网关的当前通信状态和历史异常信息。

2.4.3.7 大屏可视化

针对配电站设备监测，本文结合 Echarts、Bmap 库设计了一款大屏可视化界面，其 Web 端界面如图 2.29 所示。主要涵盖地图显示、设备状态统计、报警类型统计、SOE 事件概况、发电量周报表等模块。

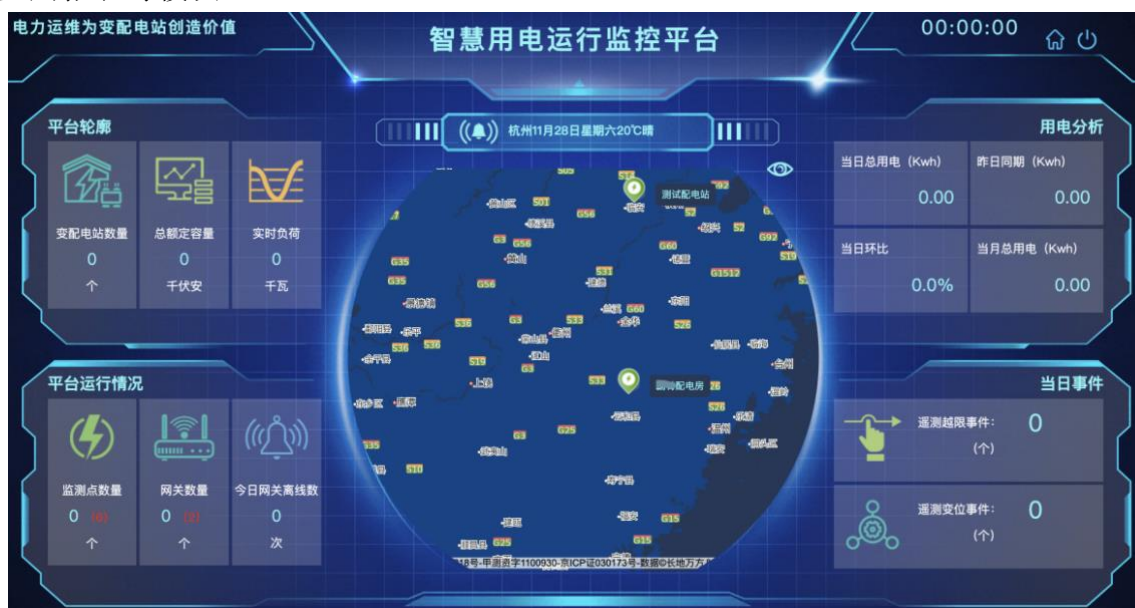


图 2.29 大屏可视化界面

2.4.3.8 变电站状态查看

包括变电站概况、运行状态、用电概括等功能模块，界面实现效果如图 2.30 所示。



图 2.30 变电站状态界面

2.4.3.9 运行环境监测

针对变电站运行环境，设计温湿度监测、线缆温度监测、仪表通讯状态查看、网关通信状态查看等功能模块，实现效果如图 2.31、图 2.32、图 2.33 所示。



图 2.31 环境监测界面



图 2.32 线缆温度监测界面



图 2.33 仪表通信状态监测界面

2.5 本章小结

本章针对变电站在线监测系统，首先对系统项目需求进行总体分析，提出了一套物联网整体设计方案，分层次对整个智能监测系统的智能网关、云平台、用户客户端等分别做了详细的设计方案介绍和实现结果阐述，在本章最后对上述各个模块的实现成果进行展示。

本文设计的变电站在线监测系统有如下优势：

(1) 设备数据状态可实时传输，智能网关扩展性强，支持多种通信接口。

(2) 稳定可靠的服务器平台支撑。设备云和自主搭建的业务云服务器集群各司其职，引入负载均衡机制实现服务器平台的高性能、高可用。

(3) 完善的软件服务。Web 监控平台集成了大屏可视化平台、组态配电图监控平台、设备用户管理等多模块，可实时有效监测变电站系统运行状况；微信小程序支持随时随地查看设备数据、接收告警通知。

第三章 基于 MGWO 卡尔曼滤波算法的故障预测方案

故障预测主要是根据系统设备历史状态数据, 预测其变化趋势, 提前确定系统可能出现故障的时间, 这种方式可以留给维护人员更多的时间去应对突发事件, 降低故障引起的损失。

目前已有不少针对电力设备故障预测的研究, 李华忠通过标准卡尔曼滤波算法实现电缆接头温度的预测^[22], 但由于采用人工设定卡尔曼滤波算法系统初始参数, 导致算法精度低、收敛性差, 影响了温度预测准确性; 文献^[23]将卡尔曼滤波与 SVM 相结合实现了风力发电系统的故障预测, 首先利用 SVM 回归建立了建立非线性状态模型, 通过卡尔曼滤波递归对设备进行预测, 提高了预测的精度, 但该算法构建模型复杂, 且运算速度较慢; 黄文群采用粒子滤波算法, 预测补偿电容容值下降程度, 实现了对设备剩余使用寿命的预测^[24]。张文新利用 BP 神经网络实现了电力变压器温升的预测^[25]。文莹等人针对 BP 神经网络在预测精度上存在的误差, 通过粗糙集理论的交叉连接方式来改进 BP 神经网络中的神经元结构, 达到跟踪预测物理设备的特征参数的目的^[26]。郑含博等人将改进粒子群算法与最小二乘支持向量机结合, 以达到预测电力变压器故障的目的, 提高了故障诊断的准确率^[27]。

目前在变电站设备故障预测上, 有通过神经网络、支持向量机、粒子滤波等技术对故障进行预测的研究, 但是此类方法模型构建复杂、特征提取困难、计算代价昂贵, 未得到较大的应用。而现阶段的设备短期故障预测方案, 主要采用以数据驱动为基础的卡尔曼滤波算法, 这种方法的核心原理是以变电站系统的运行状态参数为基础, 自回归推演出当前和未来一段时间之内的健康状态, 特点是算法简洁清晰, 无需完全掌握目标系统的专业基础知识, 模型构建简单, 算法运行效率高且性能优异。

虽然基于卡尔曼滤波算法的故障预测有一定的优势, 但是还有待改进, 传统的卡尔曼滤波算法假定噪声矩阵 Q 、观测噪声矩阵 R 两个系统初始状态值是恒定的, 通常采用人为设定的方式, 但在实际问题的求解中, 噪声参数是一个不确定的因素, 很难确定系统初始状态值, 故一般给定的初始值通常是一个近似的固定值, 在滤波过程中, 不能够自适应调整来弱化滤波过程中的误差, 降低了算法的预估精度。

根据以上分析, 本文首先针对传统 GWO 算法全局勘探能力和局部开采能力不平衡的问题, 提出一种非线性收敛因子和记忆位置更新方程双策略来提高传统 GWO 算法的全局寻优能力, 形成改进的 GWO 算法(MGWO 算法); 运用 MGWO 算法对影响卡尔曼滤波算法预估精度的初始噪声矩阵参数进行动态实时优化调整, 进而提高卡尔曼滤波算法的状态预估精度, 保障故障预测的准确性, 为工作人员提供更多应对系统突发事件的时间。

3.1 整体设计思路

故障预测整体设计流程如图 3.1 所示。

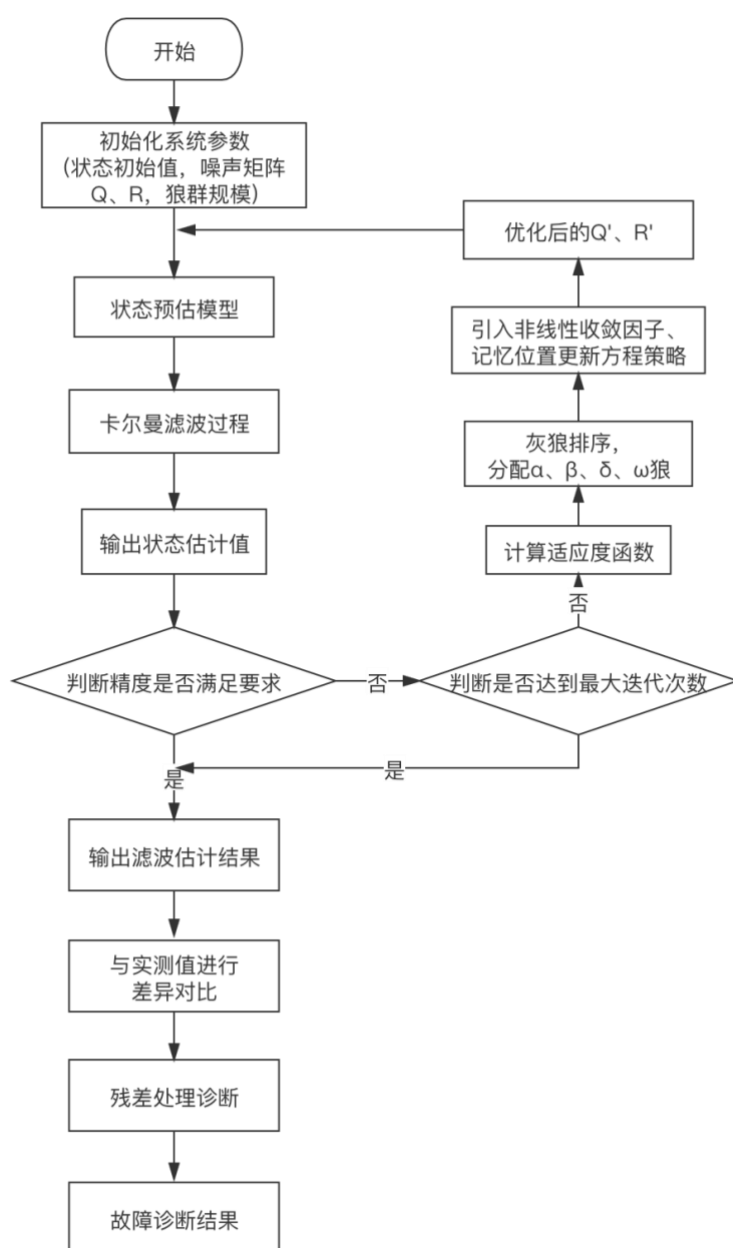


图 3.1 基于 MGWO 卡尔曼滤波故障预测整体设计流程

具体设计思路如下：

(1) 引入非线性收敛因子策略、记忆位置更新方程策略改进灰狼优化算法(Gray Wolf Optimizer, GWO)，提高 GWO 算法收敛性，形成多策略灰狼优化算法(Multi Strategy Gray Wolf Optimizer, MGWO)。

(2) 通过 MGWO 不断迭代求出卡尔曼滤波算法的系统最佳初始参数（即噪声矩阵），提

高卡尔曼滤波算法估计精度。

- (3) 利用卡尔曼滤波算法求出预测值，并与实测值进行差异对比。
- (4) 通过残差处理判断设备是否发生故障。

3.2 卡尔曼滤波原理及误差分析

R.E.Kalman 在 1960 年提出卡尔曼滤波算法，利用递推关系式来最优化估计系统状态，被广泛应用于通信、设备故障预测以及信息聚合处理等多领域^[28]。

卡尔曼滤波算法系统模型可用如下的状态空间方程来描述：

$$x_k = f(x_{k-1}) + w_k \quad (3.1)$$

$$z_k = g(x_k) + v_k \quad (3.2)$$

式(3.1)为系统状态方程， k 为离散时间， f 为 k 时刻的状态值 x_k 的转移函数， x 表示卡尔曼滤波算法中估计的状态参量， w 代表过程噪声，其均值和方差可用式(3.3)表示；式(3.2)为系统测量方程， g 为 k 时刻的测量值 z_k 的转移函数， v 代表测量噪声，其均值和方差可用式(3.4)表示^[29]：

$$E(w_k) = 0, \quad E(w_k w_k^T) = Q_k \quad (3.3)$$

$$E(v_k) = 0, \quad E(v_k v_k^T) = R_k \quad (3.4)$$

其中，过程噪声协方差矩阵用 Q_k 来表示，测量噪声协方差矩阵用 R_k 来表示。

根据状态方程和测量方程，通过标准卡尔曼滤波算法框架，就可以对状态参量进行估计。算法可以概括为两个步骤：预测阶段和更新阶段。而其中预测阶段负责求解预测状态值和状态误差协方差矩阵预测值；更新阶段负责求解滤波增益，并更新预测状态值以及状态误差协方差矩阵。具体算法步骤流程如下：

1. 系统初始化状态初始值、 Q 、 R 。
2. 标准卡尔曼滤波流程：

$$\begin{cases} \hat{X}(t_{i/k-1}) = \Phi(t_k, t_{k-1})\hat{X}(t_{k-1}) - W(t_k, t_{k-1}) \\ P_{k/k-1} = \Phi(t_k, t_{k-1})P_{k-1/k-1}\Phi^T(t_k, t_{k-1}) + Q \\ K_k = P_{k/k-1}H_k^T(H_k P_{k/k-1}H_k^T + R)^{-1} \\ \hat{X}(t_k) = \hat{X}_{k/k-1} + K_k\{Z_k - h_i[\hat{X}(t_{k/k-1})]\} \\ P_k = (I - K_k H_k)P_{k/k-1} \end{cases} \quad (3.5)$$

3. 求得滤波值。

在处理解决很多问题时，卡尔曼滤波算法的性能都很优异、效率非常高效，但是仍然有着自己的不足之处，在将卡尔曼滤波算法运用到实际工程项目中，需要有针对性的对其优化改进，主要是通过与其他算法进行有机结合或者通过调节标准卡尔曼滤波算法的系统参数等方式，以适应自身项目需求。

(1) 实际的工程环境会严重限制标准卡尔曼滤波算法中的噪声统计特性以及状态方程，不同环境的噪声不同，难以得到有关噪声的统计特性信息，导致算法无法准确估计，精度受到了严重的影响。

(2) 卡尔曼滤波算法系统状态方程的不同，其预估误差结果的差异性很大，甚至会出现算法发散的问题。Kalman 滤波算法在整个系统中起主导作用的是卡尔曼滤波增益 K_k ，卡尔曼滤波增益 K_k 的求解的好坏将严重影响 Kalman 算法的预估精度，而卡尔曼滤波增益 K_k 主要通过过程噪声协方差矩阵、测量噪声协方差矩阵两个矩阵参数来求解，因此我们需要求解出系统噪声矩阵参数的最优值，从而降低卡尔曼滤波增益 K_k 导致的误差积累和避免算法发散的问题。

在面对实际工程问题时，系统初始值 Q 、 R 的值通常较难确定，一般是根据经验给定一个固定初始值，但是如果选取的系统初始值不恰当会影响卡尔曼滤波算法的估计精确度，无法准确估计真实值。下面将详细阐述通过改进的灰狼优化算法(记为 MGWO 算法)来动态自适应调整 Q 、 R 参数值的过程。

3.3 GWO 算法及其改进策略

本节首先介绍标准灰狼优化算法(Gray Wolf Optimizer, GWO)基本原理，通过引入非线性收敛因子和记忆位置更新方程双策略，对 GWO 算法进行优化改进，提出了双策略改进灰狼优化算法(Multi-Strategy Wolf Optimizer, MGWO)算法提高算法寻优精度。最后通过 Matlab 仿真对比测试 GWO 算法改进前后的性能差异，测试表明，MGWO 算法可以有效降低标准 GWO 算法陷入局部最优的风险，为下文卡尔曼滤波算法的初始参数最优值求取奠定了基础。

3.3.1 基本 GWO

GWO 算法是一种群体智能优化算法，通过模仿生物界灰狼群体中严格的社会等级制度和狩猎行为来搜索问题的最优解^[30]。

GWO 算法中的每个灰狼个体代表一个可行解，将群体中所有灰狼个体根据适应度函数值进行排序，得出最优解为 α 狼，负责决定捕猎行为、栖息地等；第二最优解为 β 狼，为头狼 α 的最佳继任者，服从 α 狼的领导，负责协助 α 狼做出决策；第三最优解为 δ 狼，服从 α 狼和 β 狼的领导；其他解均为 ω 狼，负责平衡种群内部关系^[31]。图 3.2 展示了灰狼等级结构图。

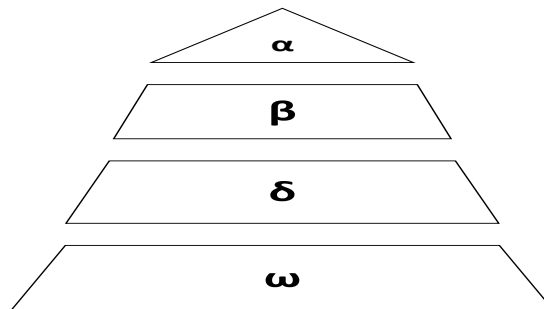


图 3.2 狼群等级结构图

狼群中适应度最好的 α 狼, β 狼, δ 狼负责指导其他的 ω 狼群, 在每次迭代过程中, 不断更新狼群位置来缩短与猎物的距离。个体位置以及个体更新方式可以用如下数学表达式表示^[32]:

$$D = |C \cdot X_p(t) - X(t)| \quad (3.6)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (3.7)$$

式(3.6)、(3.7)中, 定义猎物和灰狼位置分别为 $X_p(t)$ 、 $X(t)$, 两者之间距离用 D 表示, A 与 C 表示调节系数, 可分别用式(3.8)、式(3.9)表示^[33]:

$$A = 2a \cdot r_1 - a \quad (3.8)$$

$$C = 2 \cdot r_2 \quad (3.9)$$

式(3.8)、(3.9)中, a 代表距离协调参数, 称为收敛因子, 随着算法迭代进行由 2 线性递减至 0, 可用式(3.10)表示^[34]:

$$a = 2 - \frac{2t}{t_m} \quad (3.10)$$

式(3.10)中, t_m 为最大迭代次数。

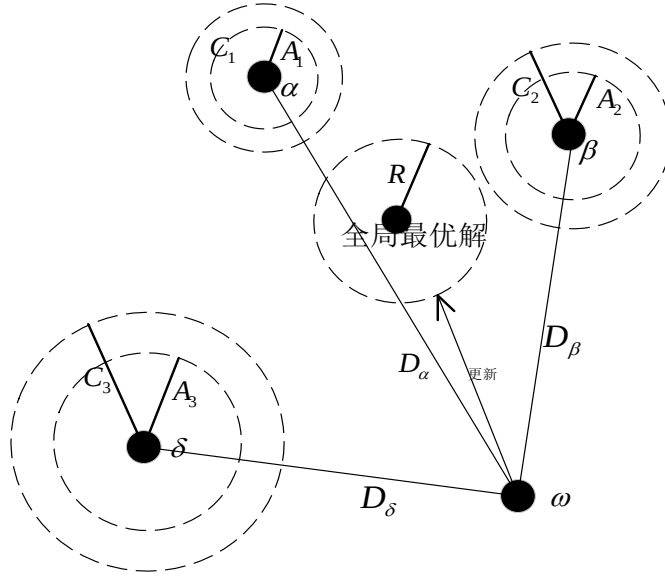


图 3.3 灰狼优化算法原理示意图

灰狼优化算法原理示意图如图 3.3 所示, α, β, δ 是潜在的最靠近猎物的三条头狼, 当灰狼探寻出猎物的大概位置时, 三条头狼带领其他狼群 ω 逐步靠近猎物, 其过程可用如下数学表达式表示:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)| \quad (3.11)$$

$$D_\beta = |C_2 \cdot X_\beta(t) - X(t)| \quad (3.12)$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X(t)| \quad (3.13)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \quad (3.14)$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \quad (3.15)$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \quad (3.16)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (3.17)$$

其中, $D_\alpha, D_\beta, D_\delta$ 分别代表 ω 狼朝 α, β, δ 狼前进的方向以及步长, X_1, X_2, X_3 表示 ω 狼针对 α, β, δ 狼前进后的位置, $X(t+1)$ 为 ω 狼当前迭代更新之后的位置。

3.3.2 非线性收敛因子策略

群体智能优化算法存在着探索和开采能力的平衡问题, 前者代表全局搜索能力, 后者代表局部开采能力。标准 GWO 算法中的收敛因子 a 是从 2 线性递减到 0 的, 然而算法在实际的搜索过程中很复杂并不是线性的。研究表明, 在迭代开始阶段, 需要增强算法勘探性, 因此要使得 a 缓慢减小; 而在算法迭代后期阶段, 为增强算法开采性, 需要加快 a 的减小速度^[35]。但实际上, 算法易陷入局部最优的缺点导致在寻得某个最优值时, 就加速了算法收敛过程, 此时寻得的“最优值”只是局部最优值, 所以在跳出局部最优后, 需要适当减缓 a 的减小速度以增强算法全局搜索能力。显然文献^[35]在跳出局部最优后继续加速 a 的减小速度, 不能充分发挥 GWO 算法的寻优性能。因此, 本文提出了一种基于改进平均适应度值的非线性收敛因子策略:

$$a = 2 - 2 \frac{e^x - 1}{e^k - 1} \quad (3.18)$$

$$k = \frac{J_{avg}(t+1) - J_{avg}(t)}{J_{avg}(t)} \quad (3.19)$$

$$J_{avg}(t) = \frac{w_1 J_1 + w_2 J_2 + w_3 J_3}{3} \quad (3.20)$$

$$w_i = \frac{X_i}{X_1 + X_2 + X_3} \quad (3.21)$$

上式中, t 表示当前迭代次数, k 为适应度调节系数, J_{avg} 表示平均适应度值, w_i 为适应度权重值。当平均适应度值变化率大时, 算法处于探索阶段; 当平均适应度值变化率小时处于开采阶段。

3.3.3 记忆位置更新方程策略

闵可夫斯基不等式 (Minkowski Inequality) 由赫尔曼·闵可夫斯基提出, 广泛应用于数学、物理、向量空间领域^[36]。文献^[37]利用明可夫斯基距离函数对模糊 c-均值算法的性能进行了改进研究。因此, 明可夫斯基距离是一种简明的参数化距离, 通过调整明可夫斯基参数 p 可以适应任何应用场景。

本文采用更广义的 Minkowski 平均数来替代基本 GWO 中普通的数学平均数, 并辅以记忆个体最优机制来调节灰狼位置更新方程, 广义的 Minkowski 平均数如式(3.22)^[38]:

$$Y_{(mm)} = \left(\frac{Y_1^{(p)} + Y_2^{(p)} + Y_3^{(p)} + \dots + Y_n^{(p)}}{n} \right)^{\left(\frac{1}{p}\right)} \quad (3.22)$$

式(3.22)中, Y 表示 Minkowski 平均数, n 表示问题维度, p 为一个常数, Y_1, Y_2, Y_3 分别对应 GWO 算法中的 α, β, δ 狼。则 GWO 中位置更新方程可更新为式(3.23):

$$\vec{X}(t+1) = \left(\frac{\vec{X}_1^{(p)} + \vec{X}_2^{(p)} + \vec{X}_3^{(p)}}{3} \right)^{\left(\frac{1}{p}\right)} \quad (3.23)$$

当 $p = 1$ 时, 式(3.23)为算术平均数, 文献^[39]论证了当 $p = -4$ 时, GWO 算法性能优化效果较好。

本文受粒子群优化算法在更新过程中记忆粒子个体最优解思想的启发, 改进标准 GWO 算法的记忆个体功能, 并结合上述 Minkowski 平均数, 提出了一种新的位置更新方程:

$$X_i(t+1) = \left(\frac{X_{i,1}^{(p)}(t) + X_{i,2}^{(p)}(t) + X_{i,3}^{(p)}(t)}{3} \right)^{\left(\frac{1}{p}\right)} + b_1 \cdot r_3 \cdot (P_{i,best}^d(t) - X_i^d(t)) + b_2 \cdot r_4 \cdot (X_j^d(t) - X_i^d(t)) \quad (3.24)$$

上式(3.24)中: $b_1 \in [0,1]$ 表示个体记忆系数, $b_2 \in [0,1]$ 表示协商系数, r_3, r_4 为 $[0,1]$ 之间的随机数, $P_{i,best}^d(t)$ 表示 i 狼个体进化史中的最优位置, X_j 表示随机选择的个体并且 $j \neq i$, 通过调节 p, b_1 和 b_2 的值, 可以平衡 GWO 算法的探索和开采能力。

3.3.4 GWO 算法优化测试

为测试 MGMO 算法与标准 GWO 算法的性能差异, 选取四个标准测试函数进行 MATLAB 仿真测试, 其公式、自变量范围、极值、维数如下表 3.1 所示。算法种群规模设为 30, 最大迭代次数设为 1000, 算法独立运行次数设为 30。

表 3.1 测试函数公式及性质

函数	自变量范围	最小值	维数
$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100,100]$	0	10
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100,100]$	0	10
$f_3(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp\left(\frac{1}{\pi} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]$	0	10
$f_4 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]$	0	10

取算法运行最后一次结果的适应度函数值绘制成迭代曲线, 以说明 MGWO 算法对比 GWO 算法的性能优越性, 结果如下图 3.4 所示。

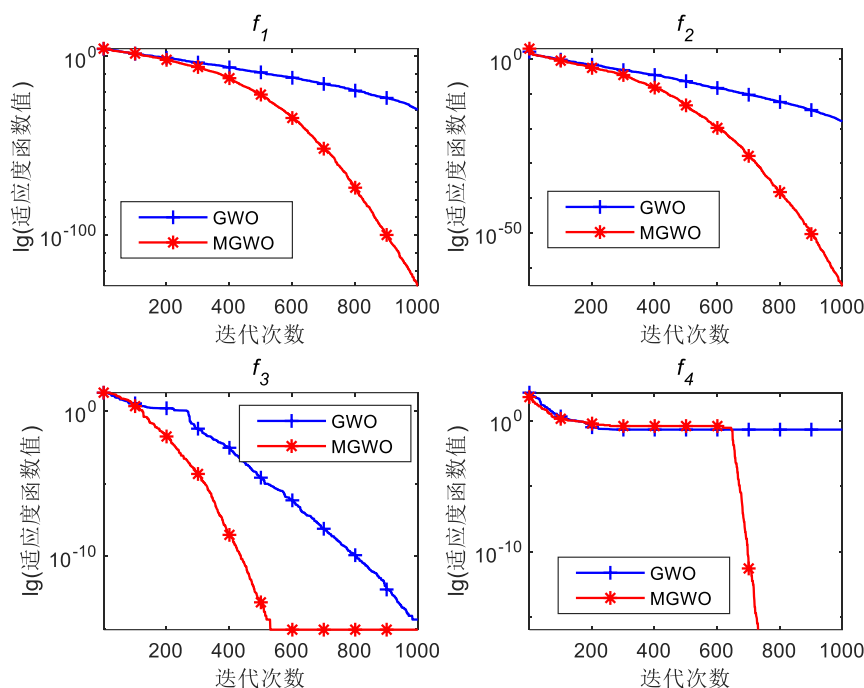


图 3.4 GWO 与 MGWO 算法收敛曲线对比

从图 3.4 可以看出，对于函数 f_3, f_4 ，MGWO 的迭代曲线都在 GWO 下方，尤其是 f_4 出现了明显的断崖式速降现象，因此改进后的算法能够有效降低标准算法容易陷入局部最优的风险。

记录算法运行结果的最优值、平均值和标准差，如表 3.2：

表 3.2 GWO 与 MGWO 算法测试结果对比

函数	算法	最优值	平均值	标准差
f_1	GWO	1.8E-32	2.23E-30	4.64E-30
	MGWO	2.3E-129	3.5E-115	1.9E-114
f_2	GWO	3.19E-20	0.33	1.8
	MGWO	1.05E-70	1.93E-64	6.67E-64
f_3	GWO	4.44E-15	0.067	0.36
	MGWO	8.88E-16	8.88E-16	0
f_4	GWO	0.025	0.16	0.11
	MGWO	0	2.42E-15	1.3E-14

从表 3.2 可以看出，对于函数 f_1, f_2 ，MGWO 算法的最优值精度分别高于 GWO 算法 97 和 50 个数量级，并且得到的标准差、平均值也都远远小于 GWO 算法。

3.4 基于 MGWO 的卡尔曼滤波算法优化设计

采用改进的灰狼优化算法对系统过程噪声协方差矩阵 Q 、测量噪声协方差矩阵 R 进行最

优值求解。将每个灰狼视作一组协方差参数进行迭代计算，记录每次迭代过程中的系统输出和适应度值，并根据适应度值进行灰狼个体排序，选出最优个体，即最优解，判断是否已达到最大迭代次数或精度满足要求，如果是则算法运行结束，输出全局最优灰狼个体作为噪声矩阵参数，本文针对卡尔曼算法的优化过程就是在每个采样周期内对噪声矩阵参数不断调优的过程。

3.4.1 适应度函数选取

适应度函数的选取合适与否决定了算法优化结果的好坏。本文在将该算法具体应用到实际工程之前，通过 Matlab 仿真验证该算法的优劣性。

以均方误差为适应度函数，其核心思想是首先计算每次算法迭代过程中的结果值与真实值的差值，之后对每个差值进行累加求和，当和最小时，即认为算法预估误差最小，性能最好。表达式如式(3.25)所示：

$$\text{fitness} = \frac{1}{n} \sum (\hat{s} - s)^2 \quad (3.25)$$

式(3.25)中， s 、 $\hat{s}$ 分别表示测量信号、估计信号， n 表示信号总长度。算法优劣好坏的评判标准为算法最终迭代得到的适应度值与极值的差值，其值越低，则表明算法效果越好，即噪声方差特性参数选取越合适。经过反复迭代，适应度值达到最小值，即是我们所求得的最佳噪声方差特性参数。

3.4.2 算法优化流程

卡尔曼滤波参数的选取的好坏会对卡尔曼滤波算法的预估精度造成影响，根据卡尔曼滤波算法的运算过程，运用 MGWO 算法对参数 Q 、 R 进行动态优化，过程如图 3.5 所示。

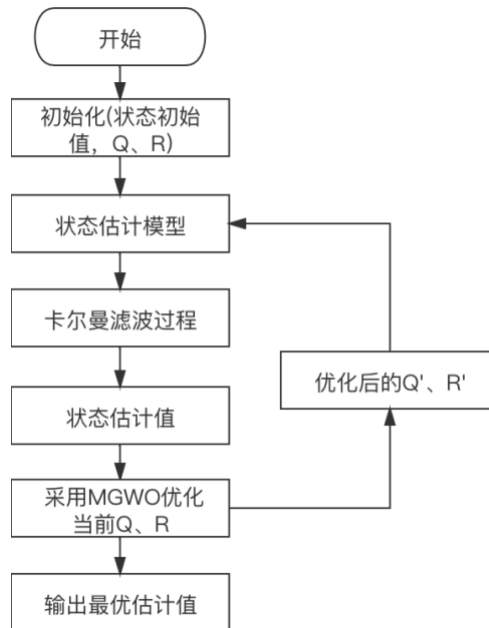


图 3.5 算法优化流程

- (1) 初始化。初始化噪声矩阵 Q 和 R ，设定狼群规模、最大迭代次数等。
- (2) 适应度函数选取为式(3.25)。
- (3) 狩猎。求解灰狼个体的适应度值，排序指定 α 狼、 β 狼、 δ 狼及 ω 狼群。
- (4) 根据上文提出的非线性收敛因子策略、记忆位置更新方程更新参数 a 、 A 、 C ，更新搜索狼群的位置。
- (5) 判断是否达到最大迭代次数或最优个体精度已满足系统要求，若是，则结束调优过程；否则重新回到步骤(2)。
- (6) 算法结束。输出全局最优灰狼个体，即卡尔曼算法的噪声矩阵 Q 、 R 最佳值。并通过滤波之后得到最终结果。

3.5 基于 MGWO 卡尔曼滤波算法的故障预测

三倍标准差可用式(3.26)表示^[40]，系统故障阈值选取为系统正常情况下的三倍预测残差标准差，如果新的预测残差明显偏离正常分布范围，则表明系统出现故障。

$$\sigma = 3 \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (3.26)$$

上式(3.26)中，数值 $x_1, x_2, x_3, \dots, x_N$ (皆为实数)，其平均值为 μ ，三倍标准差为 σ 。

下图 3.6 展示了改进的卡尔曼滤波算法的故障预测流程。

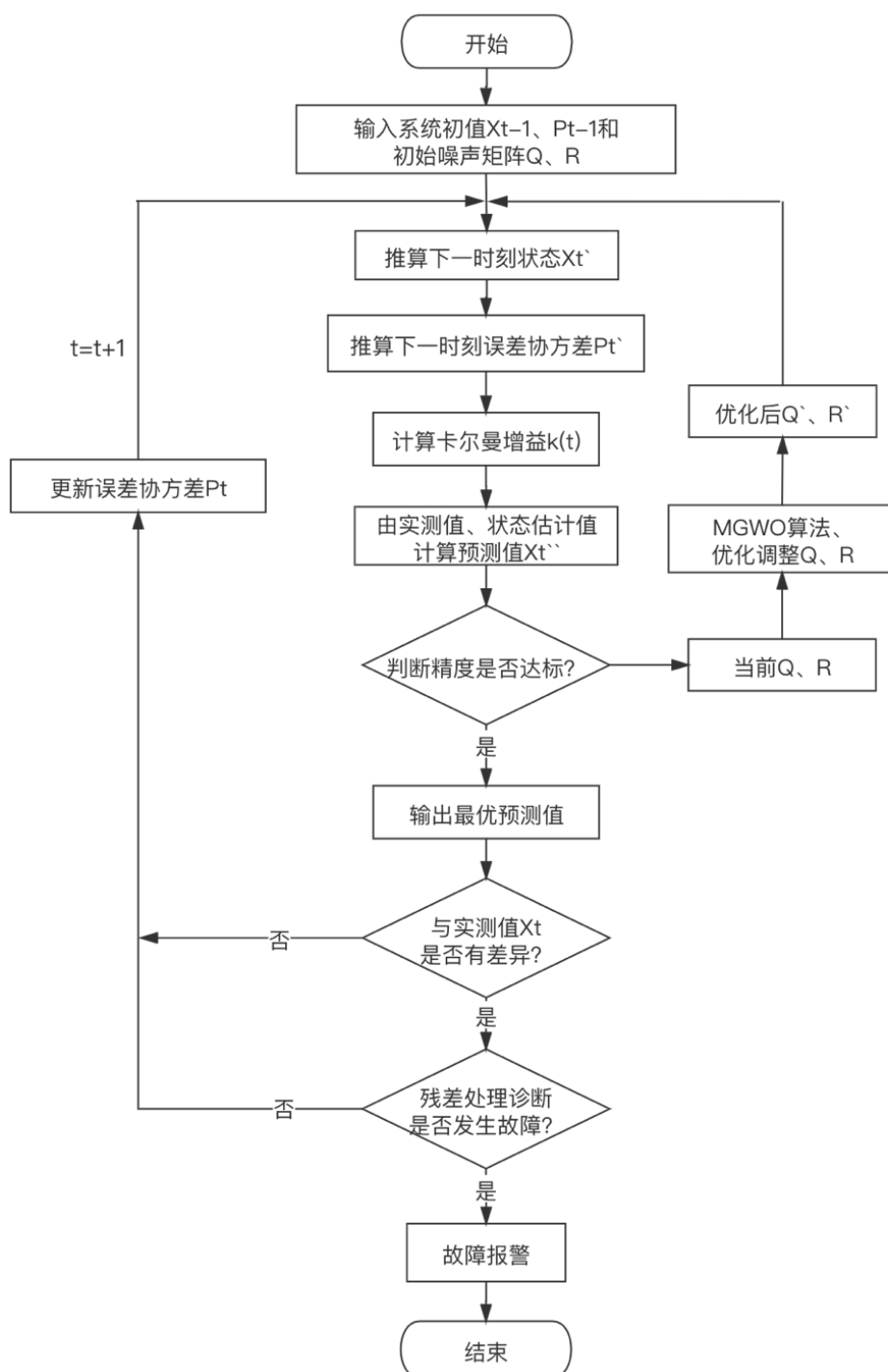


图 3.6 基于 MGWO 卡尔曼滤波算法的故障预测流程

- (1) 初始化系统初始值 x_0 、误差协方差 p_0 ，初始噪声矩阵 Q 、 R 。
- (2) 根据前一时刻的 x_{t-1} 及 p_{t-1} 求解出 t 时刻的状态估计 x_t 和协方差估计 p_t 。
- (3) 根据时间、测量方程、当前时刻数值，可求得下一时刻的协方差估计 P_t^- 和卡尔曼增益 $k(t)$ ；求出下一时刻状态估计 X_t^- 并结合观测值 Z_t ，可计算出估算值 $X(t)''$ 。同时根据 MGWO 算法对 Q 、 R 进行优化求解。

- (4) 判断与实测值 $X(t)$ 是否有差异。若没有，则进入步骤(6)；否则执行步骤(5)。
- (5) 根据残差处理诊断技术判断是否发生故障。若是，则通知系统报警，否则执行步骤(6)。
- (6) 更新误差协方差 P_t ，算法自回归继续递归计算。

3.6 实验测试

为测试卡尔曼滤波优化算法性能，进行卡尔曼算法惯用测试实验定值信号进行仿真测试。仿真环境同 3.3.4，仿真参数设定：实测值通过对固定幅值为 25 的信号加入高斯白噪声来模拟，噪声系数为 1；种群规模设为 70，算法最大迭代次数设为 120，空间维数（即噪声方差个数）为 2， Q 为 0.01， R 为 0.25。运行结果如下图 3.7 所示。

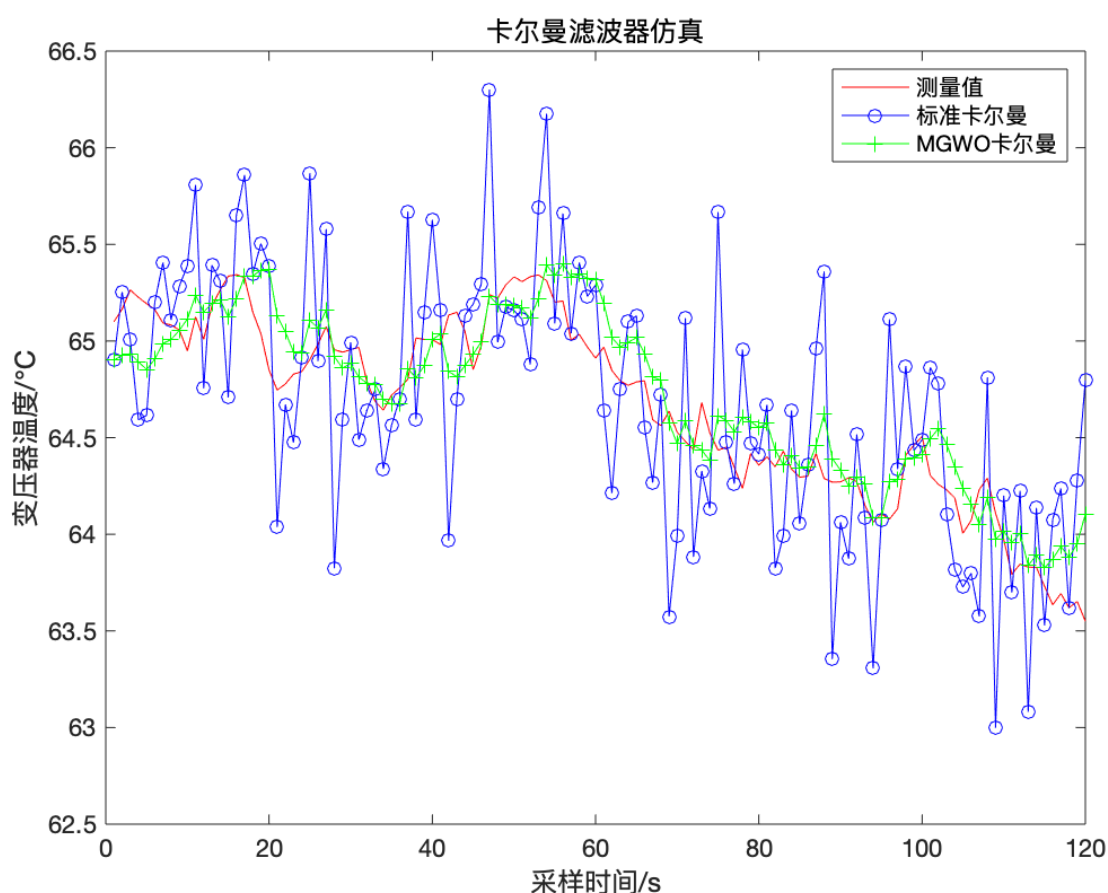


图 3.7 标准卡尔曼与 MGWO 卡尔曼算法的结果对比图

从图 3.7 可看出，由于标准卡尔曼算法 Q 、 R 是人为设定的，所以效果主要取决于初值；而 MGWO 卡尔曼算法是通过不断迭代得到最佳参数估计值，其结果要更为可信和稳定。

同时给出 Matlab 仿真误差对比图，如图 3.8 所示。

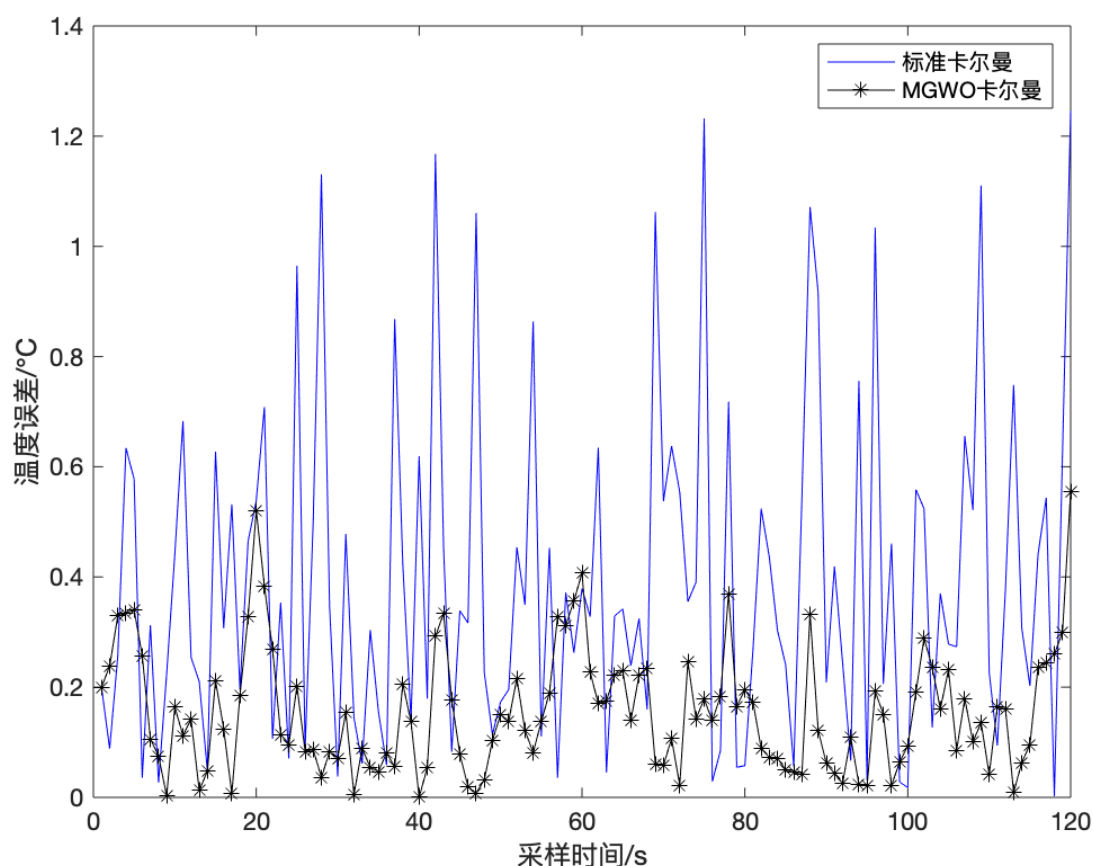


图 3.8 标准卡尔曼与 MGWO 卡尔曼算法的误差对比图

从图 3.8 可以看出，MGWO 卡尔曼算法的预估误差曲线整体更趋于 0，稳定性最高；标准卡尔曼算法稳定性最低，误差波动起伏较大。

为验证本文所提预测方法在实际应用中的性能，以电力变压器温度为例对比卡尔曼算法改进前后的性能差异。实验数据来自浙江某变电站，通过传感器每隔 10min 采集变压器温度数据，持续采集 1 天，得到 144 组温度数据，之后对每小时 6 组数据取平均数，得到 24 组 1 天的整点数据。记录卡尔曼滤波算改进前后的预测结果，如表 3.3 所示。

表 3.3 卡尔曼滤波算法改进前后的电压器温度预测结果对比

样本号	实测值/°C	标准卡尔曼算法预测值/°C	MGWO 卡尔曼算法预测值/°C
1	67.5	68.1	68.52
2	65.06	65.09	66.48
3	62.06	63.07	63.86
4	59.44	60.8	59.31
5	56.44	58.1	58.5
6	54.94	56.15	55.54
7	54	54.82	53.48

8	53.81	54.2	53.67
9	56.06	55.35	55.52
10	60	58.22	58.54
11	64.5	62.1	63.5
12	68.25	65.9	67.65
13	71.25	69.21	70.5
14	73.5	71.86	72.53
15	74.63	73.57	73.79
16	75	74.45	74.5
17	74.63	74.56	74.63
18	72.19	73.2	73.56
19	70.31	71.37	71.48
20	67.88	69.21	68.58
21	66.38	67.46	66.45
22	65.63	65.33	66.58
23	64.88	65.43	64.51
24	64.13	64.63	64.37

本文以均方根误差(RMSE)为标准衡量算法的预测精度，RMSE 计算公式如式(3.29)：

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.29)$$

式(3.29)中， y_i 代表实测值， $\hat{y}_i$ 代表预测值， n 代表样本数目，RMSE 值越小代表预测误差越小^[41]。

分别计算卡尔曼滤波算法改进前后的均方根误差(RMSE)，结果如表 3.4，结果表明 GMWO 卡尔曼算法的 RMSE 降低了 60%，预测精度较高。

表 3.4 卡尔曼滤波算法改进前后的均方根误差(RMSE)

	标准卡尔曼算法	MGWO 卡尔曼算法
RMSE	2.7911	1.2189

为更直观地反映算法改进前后的性能差异，给出预测趋势对比图，如图 3.9 所示。

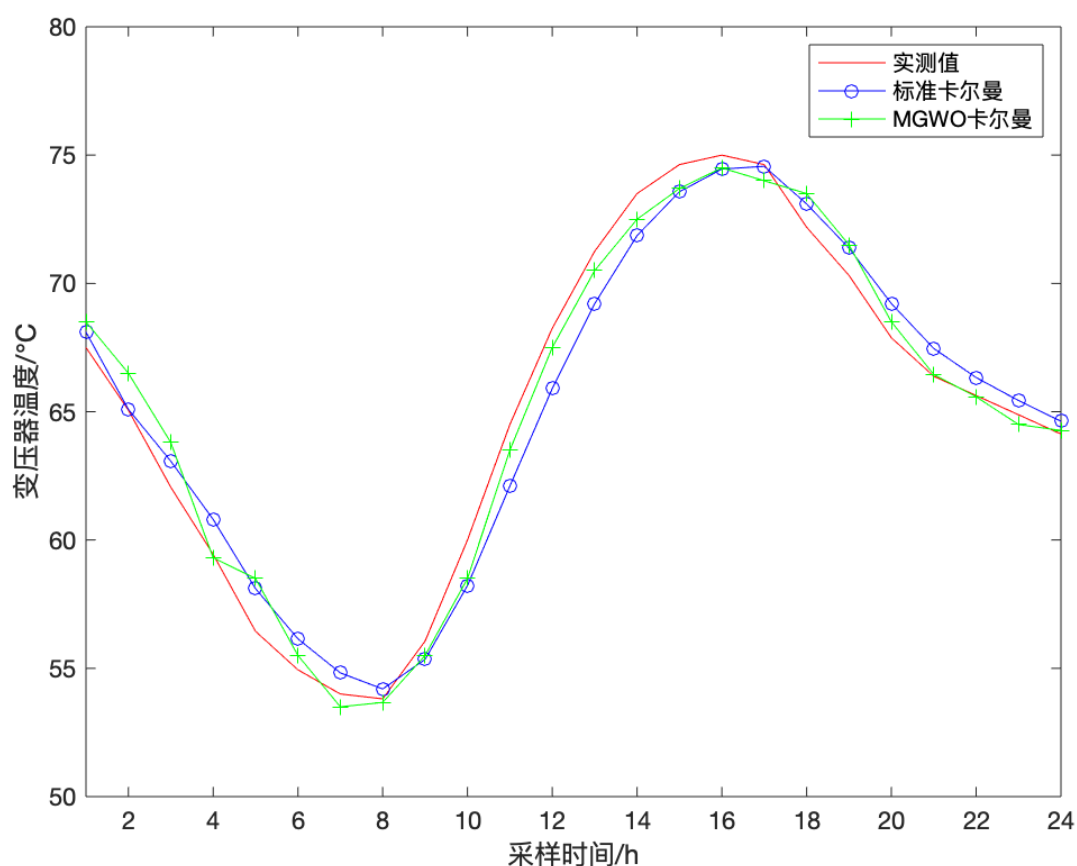


图 3.9 算法改进前后预测值与实测值吻合情况对比

图 3.9 直观反映了卡尔曼算法改进前的预测值与实测值吻合情况。在初始阶段，由于标准卡尔曼算法初始参数是依据经验人工设定的，其预测值更接近实测值；而在算法执行后期，由于 MGWO 卡尔曼算法通过不断迭代找到系统最佳初始参数，预测值曲线整体更贴合实测值曲线。经计算，预估精度的均方根误差相比标准卡尔曼算法减少了 60%，预估精度更高。

3.7 本章小结

本章首先分析了现有卡尔曼滤波算法及其他预测算法在故障预测中的应用。针对卡尔曼滤波算法在预测精度的不足，通过 MGWO 算法对其进行初值参数最优搜索，提高了卡尔曼滤波算法的预估精度。最后以变压器温度为例进行性能测试，测试表明，文本提出的基于 MGWO 的卡尔曼滤波算法的状态预测精度满足系统要求，均方根误差降低了 60%，为故障预测的准确性奠定了基础。

第四章 基于 SVG 与 WebSocket 的 Web 组态设计方法

随着变电站监测系统的逐渐智能化，组态软件也逐步成熟。组态软件具有扩展性高、直观性强等优点，在整个变电站监测系统中扮演着门户的角色，用户可以根据变电站实际电气接线，利用组态软件平台将其构建并映射到 Web 界面上，达到远程直观监测变电站运行情况的目的。

组态又称“二次开发”^[42]，其核心思想是：(1)开发人员预先提供基础图元库并部署到组态软件中；(2)用户通过组态软件根据现场设备连接情况构建成可视化图形界面，并保存发布到云端；(3)服务端与实时数据库进行通信交互，并将运算处理之后的数据实时下发至组态软件界面进行展示。最终达到降低开发成本、可视化实时监控现场设备运行状况的目的。

目前市面上有很多成熟的组态软件设计方法，其核心模块设计大致可归纳为三个角度：架构模式、图元编辑、数据通信。冷波采用 B/S（浏览器/服务器）架构模式、Ajax 通信技术设计了一套 Web 组态监控平台^[43]；张文硕结合 SVG 和 Ajax 通信技术，设计了一套基于 B/S 的监控系统，降低了软件系统开发维护成本，但由于采用 Ajax 通信技术，系统在数据实时推送方面延时较高^[44]；王彦文等直接采用第三方组态软件——力控软件设计了一套变电站在线监测系统，达到远程监测变电站系统运行状况的目的，但第三方组态软件服务费用较高、系统安全稳定性得不到保障^[45]；张华基于 C/S(客户机/服务端)架构模式设计了煤矿安全监控系统，利用了 PC 机数据运算的优异性，但跨平台兼容性差，强依赖于操作系统的版本^[46]。

经过大量实际项目验证，总结现有组态软件存在以下缺点：

(1) 开发成本高。一方面传统组态软件大多数基于第三方厂商平台开发，如组态王、Wincc 等，这些第三方软件普遍存在收费高、定制化低、开放性低的缺点；另一方面普遍采用 C/S 架构模式，选择 PC 机作为客户端载体，这要求开发人员一次性发布组态编辑软件版本，并且每次软件版本的升级迭代都需要客户支付版本升级费用。

(2) 平台兼容性低。不同于 Web 浏览器的高兼容稳定性，客户机种类多种多样，并且每一种客户机内又细分多个版本，传统 PC 上位机组态软件难以应对多平台兼容问题。除此之外，不同的组态软件厂商采用不同的图形格式，这大大降低了软件的灵活性、开放性以及跨平台兼容性。

(3) 软件界面渲染性能低。传统组态软件大多使用体积较大的位图文件，并且没有对图形元素的二次拼接采用高效合理的数据结构进行运算处理，界面渲染方式单一低效，导致界面加载速度缓慢，用户体验感差。

(4) 画面动态显示实时性低。设备状态变化和数据的实时显示是整个变电站监测系统中最重要的部分，而传统组态软件对设备数据的实时通信大多采用轮询请求的方式，难以保证设备数据、报警信息的实时传输。

本文针对以上问题，聚焦于组态软件设计的两大核心模块：组态编辑模块、组态通信模块。采用 B/S 架构模式，提出了一种稳定高效、无第三方插件、跨平台兼容性高、通信延时低的 Web 组态设计方法。针对 Web 组态编辑模块，采用 SVG(Scalable Vector Graphics 矢量图形描述语言)作为组态软件基本图元格式，并利用 SVG 的 DOM 可操作性，通过 JavaScript 将设备状态数据绑定到图元属性，达到组态界面动态显示的目的，采用有向图理论降低组合图形的多输入多输出的逻辑运算复杂程度，进而提高组合图形开发效率和增加用户构建组态界面的体验感；对于组态通信模块，采用 WebSocket 技术实现 Web 端-服务器端的双端实时通信。

通过以上组态软件两大核心模块的设计优化，不仅满足了变电站监测系统中配电图可视化构建、查看的功能需求，同时由于 SVG、WebSocket 技术的引入，提高了应用程序的跨平台、跨终端性能。除此之外，架构设计的调整不仅实现了组态软件界面展示阶段的跨平台，而且也满足了跨平台开发组态软件的需求。

4.1 基于 SVG 的 Web 组态编辑模块设计

4.1.1 Web 组态编辑模块总体设计

Web 组态软件作为整个变电站监测系统的一个核心子模块，其用户身份、权限校验由整个外围软件应用系统统一处理。用户登录成功后，可分别通过“组态编辑”、“组态查看”两个入口来选择不操作。Web 组态编辑模块的具体设计框图如下图 4.1 所示。

(1) Web 组态开发人员提前将通过 SVG 技术构建完成的多种基础静态图元、基本形状、动态控件元素封装成 JSON 格式属性描述文件，并上传到云数据库中，以使用户构建组态时调取，进行界面组态操作。

(2) 用户进入到组态编辑模块后，Web 端立即通过网络请求调取静态图元属性描述文件，并交由属性描述文件解析引擎解析还原成静态图元、图形等初始化状态，并通过浏览器 GUI 渲染线程渲染出页面。用户基于多类系统自提供的图元控件进行可视化拼接组合操作，最终完成界面组态。

(3) Web 组态编辑包含静态界面拼接、动态图元数据绑定两大模块。首先针对图元控件设计开发描述文件 JavaScript 动态脚本，并将其预先存储至后端服务器，方便以后组态系统远程调取并交由解析引擎还原为动态函数、方法，用户可基于动态函数进行数据源绑定，实现设备状态数据与组态界面的同步。

(4) 上述过程相当于对基础 SVG 图元描述文件、JavaScript 脚本进行二次开发的过程，用

户完成所需要的界面组态之后,即生成新的 SVG 图元描述文件和对应的 JavaScript 动态脚本。

(5) 工程发布。用户需要将界面组态完成后的工程文件发布到云服务器中,以实现远程查看、编辑的功能。待发布工程文件分为静态资源文件和动态资源文件两种。静态资源文件主要是网页 HTML 静态文件、SVG 属性描述文件、CSS 层叠样式文件、基础图形等;动态资源主要是 JavaScript 动态脚本。

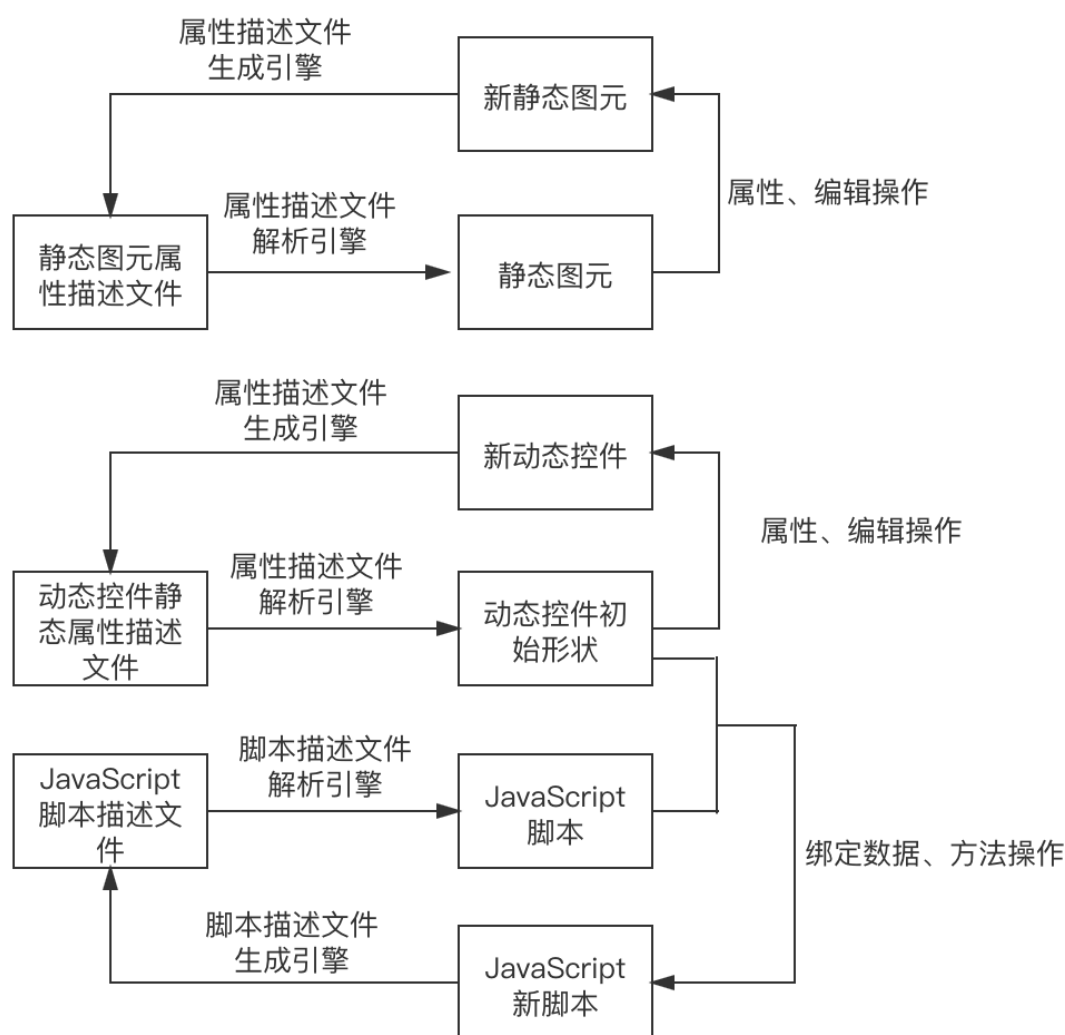


图 4.1 Web 组态编辑模块设计框图

Web 组态发布系统架构按照功能可划分为两大模块:浏览器端、服务器端。浏览器端是人机交互的窗口,可通过内部解析引擎、渲染引擎将 HTML、CSS 等格式的文件呈现到界面上,集成了组态编辑、界面管理、历史数据查看等功能模块;服务器端是数据处理中心,向上通过 SQL 语句与数据库进行数据通信,向下通过 HTTP、WebSocket 接口与浏览器进行数据交互。Web 组态发布系统架构如下图 4.2 所示。

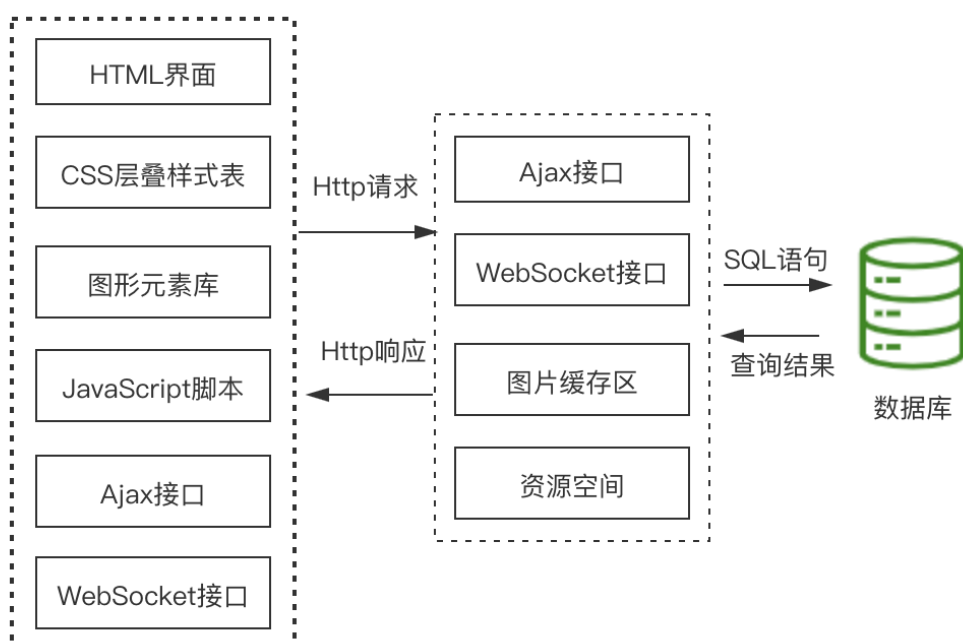


图 4.2 Web 组态发布系统架构图

4.1.2 基于 SVG 的图元编辑设计

基础图元是构建 Web 组态最基本的元素。图形元素的设计，要满足以下三点要求：

(1) 体积轻量、跨平台兼容性强。

(2) 可快速便捷组合成复杂图形。图元的设计必须具有强二次开发的特点，方便用户进行二次开发组态界面。简单图元组合成复杂图形的过程中涉及到复杂的输入输出逻辑运算，因此要采用合适的算法或数据结构来简化这些复杂的多输入多输出数据处理环节，具体将在下一小节详细阐述。

(3) 数据交互性强。图形元素不单单要展示图形，更为重要的是满足可操作交互性需求，可以接收服务器传输的数据并实时更改自身状态，方便用户更好的监测系统状态。

4.1.2.1 图元设计

SVG 是一种矢量图形描述语言，由于体积小、无失真放大、可交互性强、跨平台兼容性强等特点，被广泛应用于网络图形设计^[47]。

图元是构成组态界面所需的各种图形元素和类之间相互关系的统称^[48]，分为静态图元和动态图元两种。静态图元主要包括画面背景、监测对象的静态映射图形等一些相对不变的元素；动态图元主要负责被测对象的状态变化反映，能够将设备数据、状态变化及时反映到界面上。结合 SVG 技术特点，将图元划分为背景图片、基础矢量图元、动态可交互控件三大类，具体图元类型划分如下图 4.3 所示。

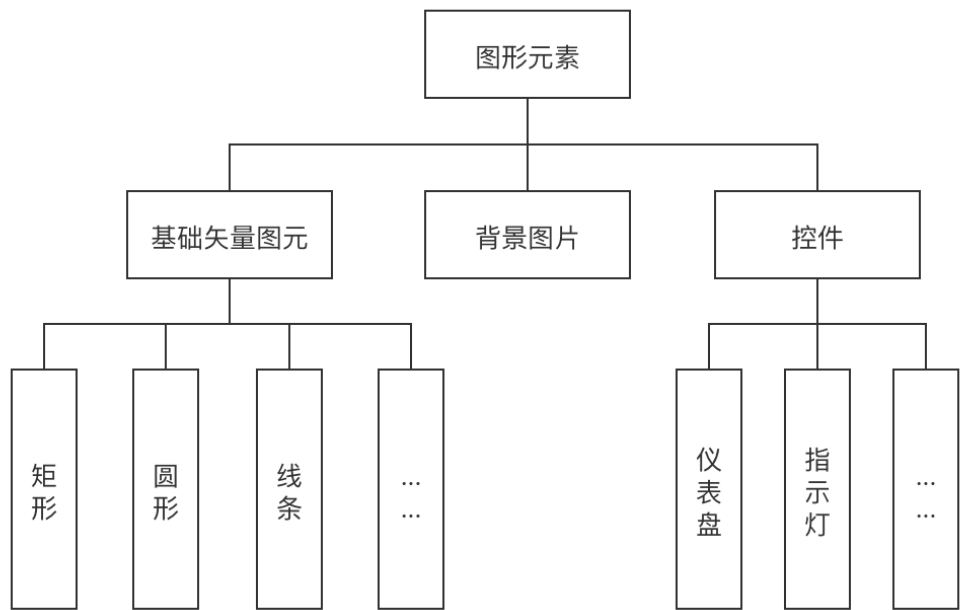


图 4.3 图元类型划分

为满足各类图元的通用性、移植性、扩展性等要求，基于 JavaScript 的原型链模型将图元对象的共同特征和公有属性抽象成类。公共基类定义为图元库的根节点 **graphic**，由所有图元特征抽象而成，其属性和方法如下表所示。其他图元类都继承于此公共基类 **graphic**，称为派生类。各派生类首先都继承有公共基类的属性和方法，按照子类需求按需调用；与此同时也设计有自身的属性和方法，当用户操作图元时，会根据图元唯一标识符 **graphicId** 判断所属类型，引入对应的属性和方法。图形类的结构层次抽象成软编码，如图 4.4 所示。

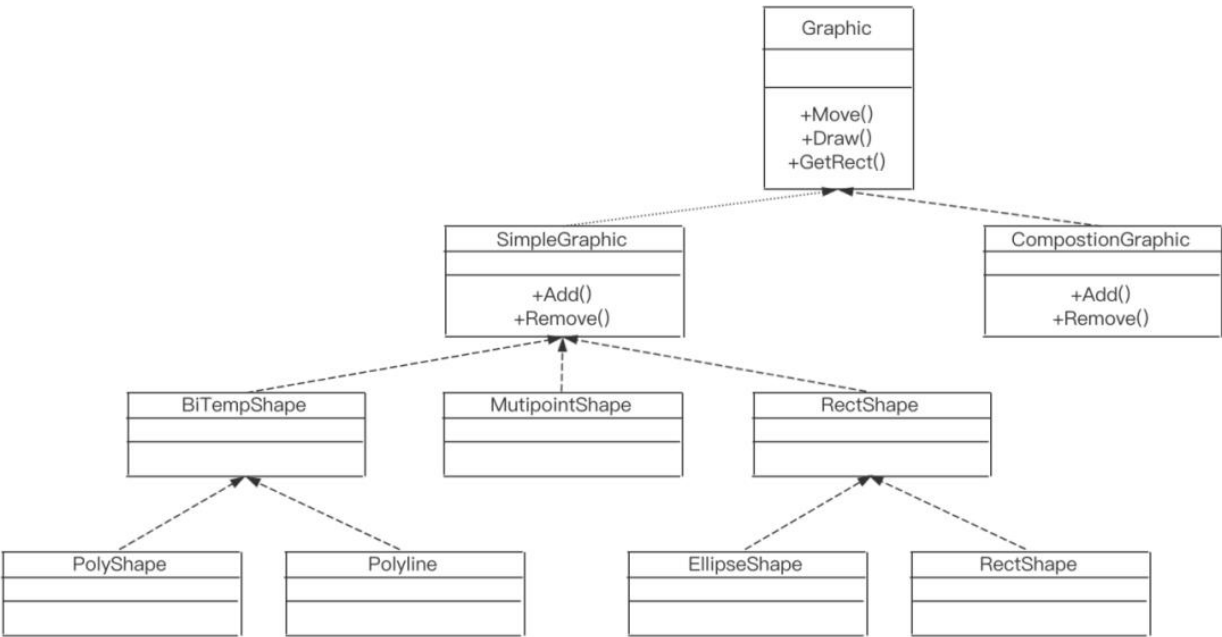


图 4.4 图元类抽象结构图

为了达到简单图元可组合复杂图形的目的，不仅需要对上述基础图元抽象成类，还需要定义其容器类来高效方便地包含这些图形。由于基础图元类与容器类的差异性，其 API 接口势必不同，这就会在组合拼接复杂图形的过程中涉及到多输入多输出的复杂运算，设计一种合理有效的图形组合算法模型显得尤为重要，具体设计过程将在 4.1.3 小节详细阐述。

4.1.2.2 图元操作设计

图元格式各式各样，而 SVG 在矢量图形绘制方面具有 DOM 可操作性、无失真放大、代码编辑简单清晰、体量小等优点。SVG 自封装了六种常见的基础图元 API 接口，如表 4.1 所示。根据这六种 API 就可以二次开发变电站监测系统中配电图所需要的图元库，利用 SVG 的 DOM 可操作性，可以对其属性值实现 JavaScript 脚本动态修改，当设备生产数据或状态发生改变时，可通过修改其 SVG 属性值实现 Web 组态界面显示的动态变化，为用户提供了良好的交互性。

表 4.1 SVG 基本形状及说明

形状	标签名称	说明
圆	<circle>	用来创建圆，并可指定圆心、半径
线条	<line>	用来创建线条
矩形	<rect>	用来创建矩形及其变种
多边形	<polygon>	创建至少包含三个边的复杂图形
折线	<polyline>	顶点在指定点，只包含直线的形状
椭圆	<ellipse>	用来创建椭圆，并可指定中心、长短轴

以 SVG 绘制椭圆图形为例说明图元绘制方法，其他图元绘制情况类似。

- (1) 引入外标签<svg>作为图元代码容器。
- (2) 通过内标签<ellipse>定义 SVG 形状为椭圆形。
- (3) 设置椭圆的中心纵坐标、横坐标。
- (4) 设置椭圆的垂直、水平半径(ry、rx 属性)。

(5) 设置 CSS(Cascading Style Sheets，层叠样式表)属性(style 属性)，可通过 JavaScript 脚本对其进行颜色、椭圆边框宽度、椭圆边框颜色的动态更改。

4.1.3 基于有向图的组合模型算法设计及实现

简单图元构建完成之后，需要将其拼接组合成复杂图形以映射到实际变电站监测系统配电图中。而传统组态软件只是采取将简单图元类进行分类放置到 Web 界面中的方式，没有考虑到对其进行组合模型算法的设计，导致用户在界面组态的过程中必须要对简单图元类或者容器类进行区分，这大大降低了 Web 组态软件的灵活性、简单易操作性。因此，有必要设计一种能够处理多输入多输出的组合模型算法，使得多个简单图元能够简单地组合成复杂图形。

4.1.3.1 组合模型的算法研究

Web 组态软件中图元的数据交互处理过程，可抽象为输入、数据模型运算、输出三个模块。对于某个数据处理过程，其输入变量可以为一个或多个，经过数据模型运算处理之后，输出量也可能为一或多个，需要有效规划输入输出之间的逻辑关系，以便能够简单地处理复杂图形多输入量多输出量的逻辑关系。

对于单输入、单输出的简单图元处理情况，可以采用简单的链表数据结构来表示，前一节点处理得到的输出量即为后一节点的输入量，其输入量、输出量的映射关系简单清晰；而对于多输入量、多输出量的组合拼接的复杂图形处理情况，其输入量、输出量不是一一对应的，所以数据处理过程复杂，而利用本文提出的组合模型算法就可以将复杂图形简单化，从而设计出 Web 组态软件中的常用组合图形。本文在充分研究计算机科学几种数据结构的基础上，将运算环节的组合模型抽象为有向图数据结构。向图结构如下图 4.5 所示。

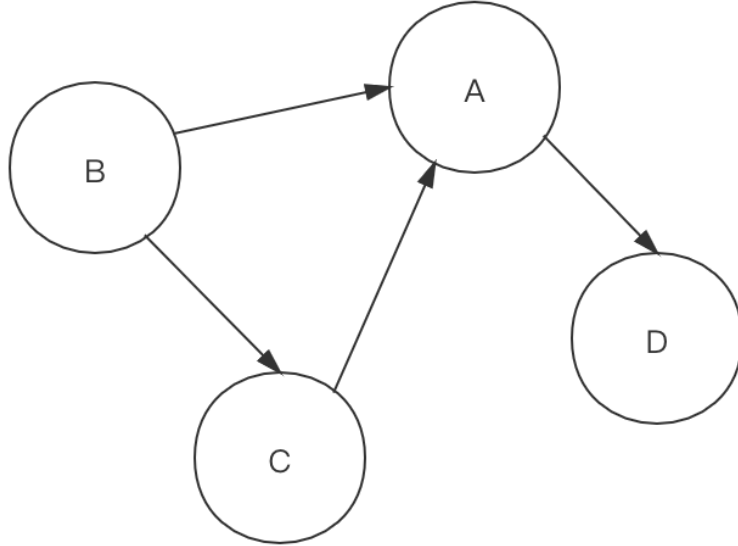


图 4.5 有向图结构示意图

上图 4.5 中，任意两个顶点之间的边均为有向边。用弧 $\langle A, D \rangle$ 来表示连接顶点 A 到 D 的有向边，A 称为弧尾，D 称为弧头。有向图可以简单理解为有序的二元组 $G(V(G), E(G))$ ^[49]：

(1) $V(G)$ 表示由 G 的顶点构成的一个有限非空集合。

(2) $E(G)$ 表示由有向边(directed edge)构成的一个 $V * V$ 的子集（有序对），即弧集^[50]。边 $e = (v, w)$ 表示从顶点 v 出发，以符号 $A(v)$ 表示从顶点 v 出发的所有边，即 $A(v) = \{(v_0, v_1) \in E: v_0 = v_1\}$ 。

有向图 $G(V(G), E(G))$ 的路径可表示为： $p = \{v_1, v_2, \dots, v_k\}$ ，其中 $v_i \in V, (1 < i < k)$ ，且 $(v_i, v_{i+1}) \in E (1 < i < k)$ ，即一个非空的顶点序列。顶点 v_{i-1} 表示顶点 v_i 的前驱，路径 p 中除首个元素外的每个元素都有一个前驱，顶点 v_{i+1} 表示顶点 v_i 的后继，路径 p 中除末尾元素外的

每个元素都有一个后继。用回路来表示头顶点到尾顶点相同的路径，即 $v_i = v_k$ ，其长度为 1，路径形如 $\{v, v\}^{[51]}$ 。

4.1.3.2 基于有向图的组合模型算法实现

借鉴软件组合模式(Composite Pattern)的思想，将图元类的结构进一步抽象为组合图形类结构图，如图 4.6 所示。

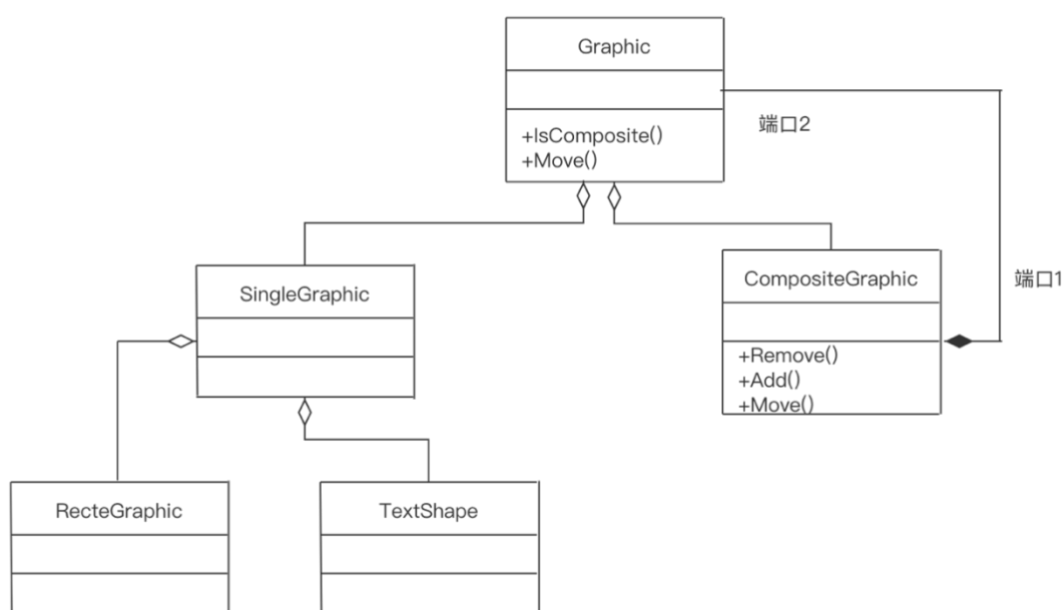


图 4.6 组合图形类的结构层次图

上图 4.6 中，Graphic 代表公共基类，是根据子类公有属性、方法抽象得出，例如 Componentposite-graphic 类是它的一个子类，利用接口的一致性，在包含基类 Graphic 接口的类都能直接调用 Componentposite-graphic。我们将添加、删除、移动等公共接口都交由 Componentposite-graphic 类进行统一协调管理。

通过基础图元构建复合图形时，由于多输入量多输出量的复杂逻辑关系，涉及到复杂的逻辑运算，这就使得系统设计人员难以把控所有的输入输出关系。有向图数据结构在处理复合图形的多输入量多输出量对应关系时有着先天优势，可以将其作为运算环节的组合格式结构。具体算法步骤如下：

(1) 判断运算环节的有向图是否有环。在前文有向图理论分析中，有向图在某些情况下是能够产生环路的，而在 Web 组态系统中简单图元组合成复杂图形的实际运算过程是不可能出现环的，即多输入量多输出量之间产生回路。因此，必须将组合格式简化为有向无环图。通过深度优先搜索算法(DFS)遍历运算环节构成的图，若在遍历过程中发现反向边则判定图中有环。

(2) 处理平行边情况。实际系统中会出现多输出量的情况，会导致组态过程中出现“平行边”，如图 4.7 中 a 部分所示，环节 1 的两个输出量作为环节 2 的两个输入量，而通过有向图

却不能表示这种特殊情况，所以需要消除“平行边”。

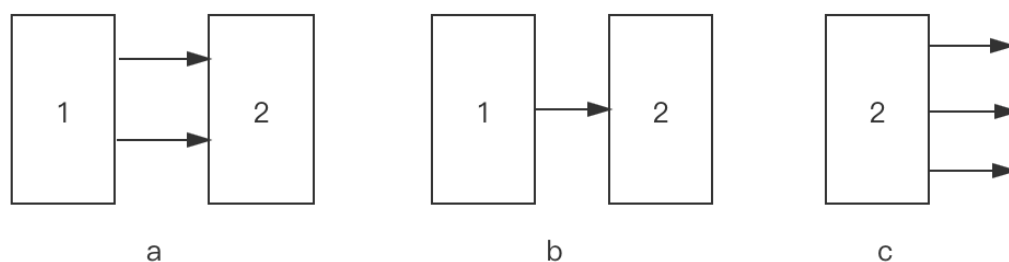


图 4.7 平行边环节

上图 4.7 中，是将输入输出参数的一对一关系定义为有向边的物理意义，所以产生了平行边情况。借鉴递归聚化思想，将前后两个环节的运算关系抽象为有向图的实际物理意义。则上图 4.7 中 a 部分代表 in-out(输入输出)关系的两条有向边，可以转变为上图 4.7 中 b 部分代表前后运算关系的一条有向边。如上图 4.7 中 c 部分所示，对于任意多输出量的起始环节，都可抽象为该环节而不是该环节的具体某个输出量。

(3) 处理环路情况。步骤(2)简化组合运算环节的方式会导致环路的出现，如图 4.8 所示。环节 1 原本是环节 2 和环节 3 的前驱节点，也是环节 3 的后继节点，若按照图 4.8 所示的前后运算流程，将陷入死循环。而实际运算过程中，由于环节 3 的输出只是在对环节组合的下一一次运算环节才会成为环节 1 的输入。

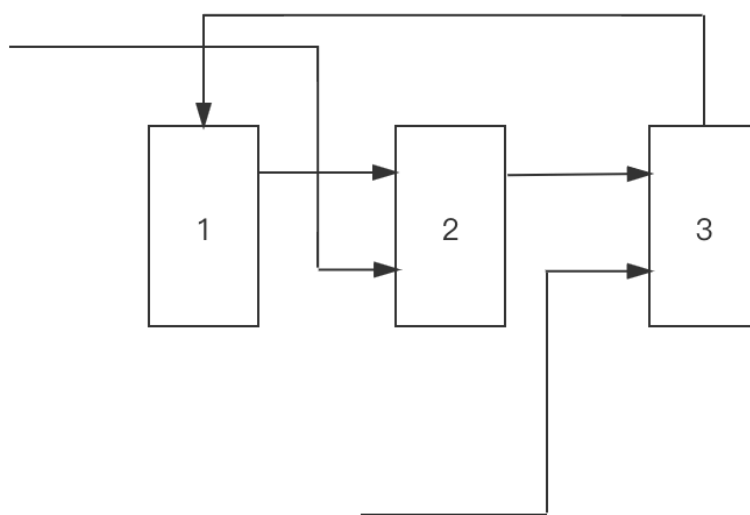


图 4.8 带有环路的组态图

为消除环路，可以在环节 3 到环节 1 之间插入源汇环节来充当特殊环节，如图 4.9 所示。图中，源环节作为始节点，当环节组合扫描过程中遇到源环节则停止扫描其前驱环节；汇环节作为末节点，当遇到汇节点时则意味着环节组合扫描结束，将停止扫描其后继环节。进行

实际环节运算时，环节 3 到环节 1 的路径中还可能出现其他中间环节，对于这种情况，亦可以通过安放一对足以切断整个环路的源汇环节即可消除回路。

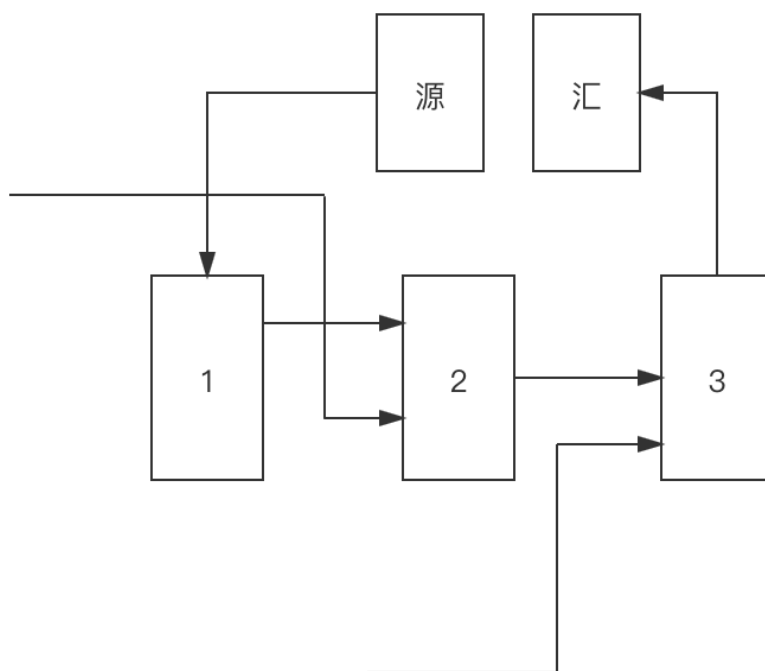


图 4.9 引入源汇节点的组态图

(4) 根据输入输出关系组合源、汇、中间三个环节，按照特定次序调取各环节的数据处理方法，继而达到控制组合模型、处理复杂数据的目的。

根据上述操作步骤最后得出 Web 组态组合模型算法，以此来消除简单图元组合成复杂图形运算过程中产生的组态环路干扰，以便高效处理组合图形控件的多输入量多输出量复杂逻辑关系。

4.2 基于 WebSocket 的 Web 组态数据通信设计

在变电站监测系统的 Web 组态中，数据量大，设备状态频繁更新，组态界面与设备数据的同步实时性要求高。因此，设计一种实时稳定的数据通信方案尤为重要。本文基于 WebSocket 通信技术设计了数据通信模块，并以 Web 浏览器作为载体实现了浏览器端-服务器端的双向实时数据推送功能。

Web 组态界面的数据展示、控件状态更改流程如下图 4.10 所示。首先服务端对智能网关上报的设备数据进行运算处理，通过前后端通信交互方案将设备生产数据或状态变更信息发送到前端，前端获取到数据后通过动态 JavaScript 脚本修改 SVG 组件属性，最终实现界面动画和实时显示效果。

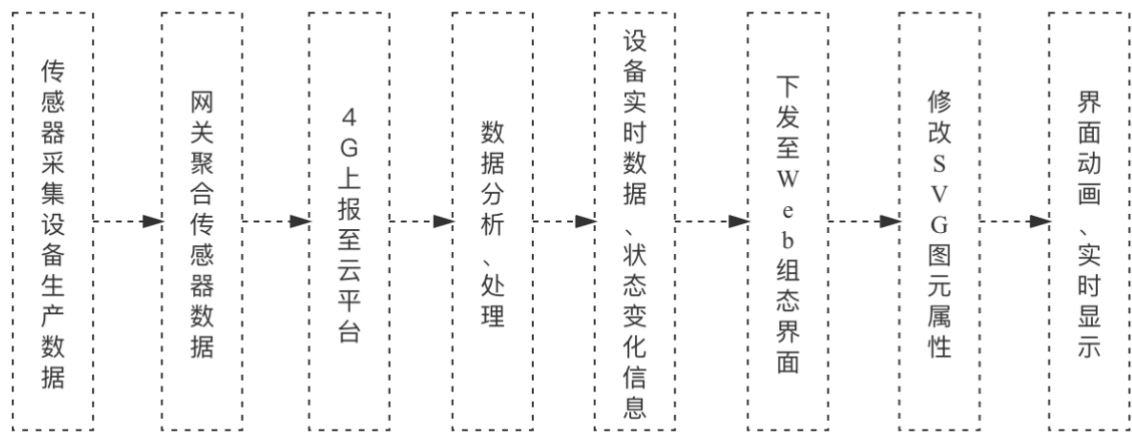


图 4.10 Web 组态通信交互流程图

4.2.1 WebSocket 概述

WebSocket 基于 TCP 协议来实现双向全双工通信^[53]。由于其稳定性强、实时性高、跨版本兼容性高的优点，已逐渐成为 Web 实时推送技术的首选技术方案，目前绝大多数浏览器已支持 WebSocket 协议。WebSocket 核心原理可以概括为：双向通信、TCP 握手。服务器与浏览器仅仅需要 TCP 握手一次，就能够建立持久性通信连接，双方都能主动朝对方传输数据^[54]。WebSocket 通信协议的握手过程如图 4.11 所示。

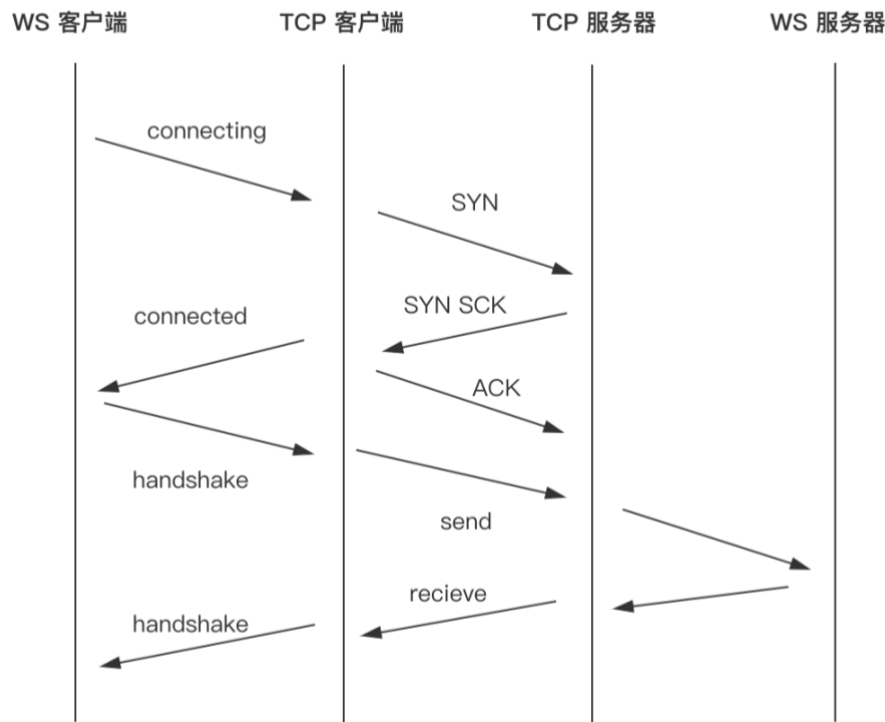


图 4.11 WebSocket 握手过程

浏览器首先向服务器请求建立基于 TCP 协议的 WebSocket 连接，如果连接建立失败，浏

览器端收到错误码信息；如果连接建立成功，浏览器将请求头中携带的版本号、主机地址等参数信息发送至服务器，服务器对其进行数据包格式校验，校验成功后将发送回复数据至浏览器，如果浏览器可以成功接收，则表示 WebSocket 连接建立成功，之后就可基于此连接进行双端的数据实时通信。

4.2.2 WebSocket 数据通信设计

服务器与浏览器端的整体通信架构如图 4.12 所示，基于 B/S 架构模式实现 Web 界面数据状态的实时变更显示。

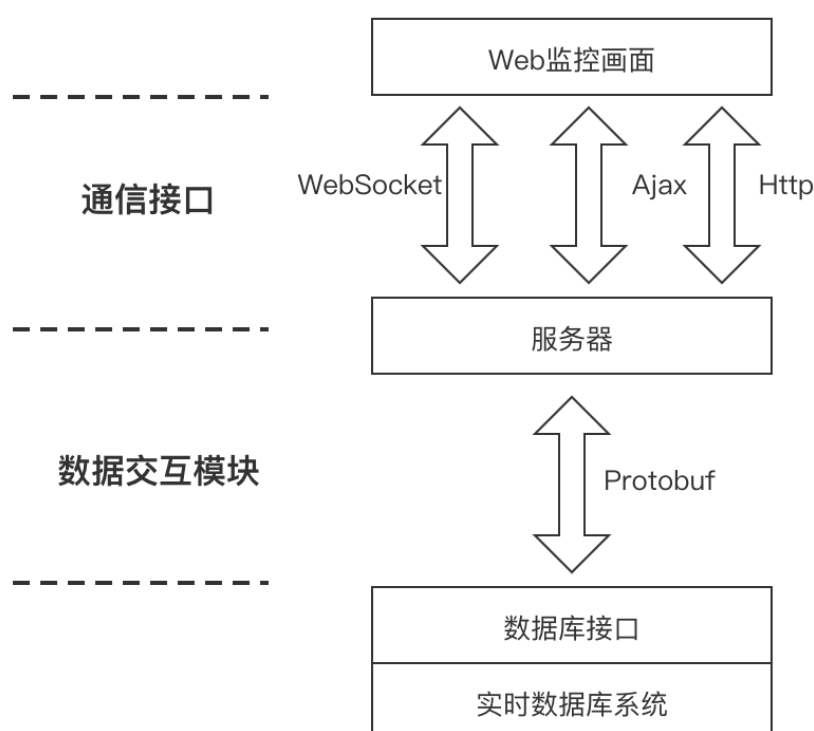


图 4.12 服务器-浏览器通信架构图

服务器一方面通过 SQL 查询语句与数据库进行数据交互，数据库中存储了两大类数据：设备数据、用户数据；另一方面通过 Http 协议、WebSocket 协议与 Web 监测界面进行数据交互。对于用户信息等实时性要求不高的一类数据，通过基于 Http 协议的 Ajax 通信技术进行浏览器-服务器的双端数据交互；而诸如设备当前状态变更、事件告警等信息数据，采用 WebSocket 通信技术，以达到实时推送数据的目的。

浏览器端与服务器端通过事件监听的方式来建立维护 WebSocket 连接，整个工作流程如图 4.13 所示。

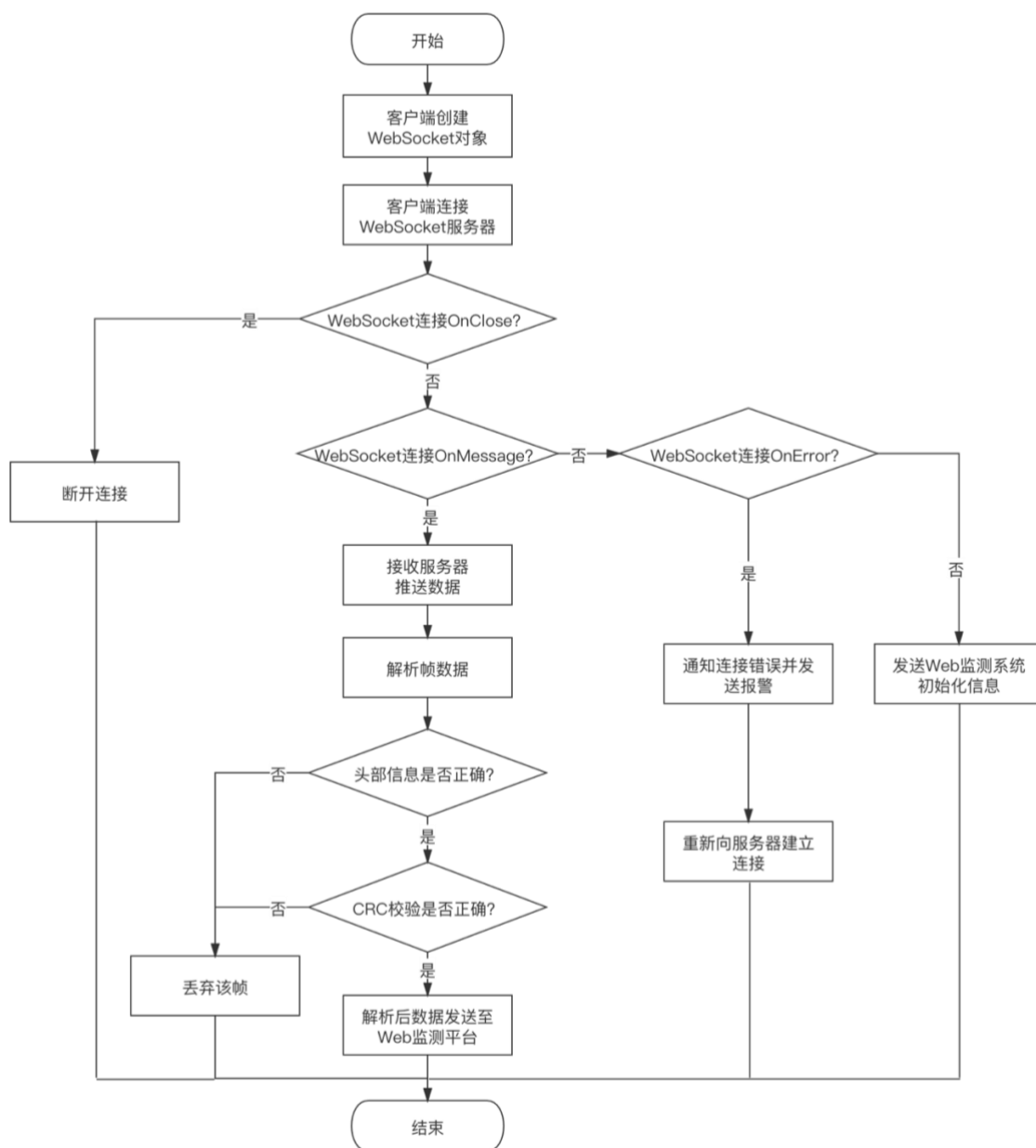


图 4.13 WebSocket 数据通信流程图

具体通信流程如下：

- (1) 浏览器端通过 JavaScript 脚本创建 WebSocket 对象，并主动发起 WebSocket 连接请求。
- (2) 判断连接是否处于 Onclose 关闭阶段，若是，则取消该连接；若不是，则转入(3)。
- (3) 判断连接是否处于 OnMessage 监听阶段，若是，则实时接收服务器端数据并执行步骤(4)。否则执行步骤(5)推送连接错误信息至浏览器端，回到步骤(1)。
- (4) 数据处理。判断状态、数据是否变更，若是，则执行步骤(6)；否则执行步骤(3)。
- (5) 判断连接是否处于 OnError 错误阶段，若是，则推送连接错误信息至浏览器端，执行步骤(1)；否则，推送浏览器端初始化信息。

(6) 将变更信息推送至浏览器端。

(7) 浏览器端接收数据状态变更信息，更新组态界面。

4.2.3 WebSocket 浏览器端实现

利用 HTML5 自定义的 WebSocket 的 API 接口，在浏览器端实现 WebSocket 通信。通信实现过程按功能划分为三大模块：连接的建立和断开、数据的发送和接收、错误信息的处理。

(1) 连接的建立和断开

创建 WebSocket 容器对象，注册 onOpen 监听回调函数，注册 onClose 监听回调函数以处理连接断开的情况。

(2) 数据的发送和接收

通过绑定 onMessage 回调函数来处理发送和接收的数据。需要强调的是，由于浏览器端-服务端数据传输格式通常为 JSON 对象，而 WebSocket 不支持 JSON 对象的透传，所以在数据发送前需要利用 JSON.stringify 方法将其转换为字符串格式。同理，在浏览器接收到服务端传输的 JSON 数据后，需通过 JSON.parse 方法将其转换为字符串格式再进行下一步处理。

(3) 错误信息的处理

通过绑定 onError 回调函数来监听 WebSocket 连接错误事件，当监听到错误时，通过界面弹窗、短信等方式告知用户数据通信异常。

4.3 实验测试

4.3.1 功能测试

Web 组态视图是由很多个组件组成，本节主要测试了视图创建、组件渲染、数据绑定、视图展示等功能，测试平台为 Opera(欧朋)浏览器，测试用例如表 4.2 所示。

表 4.2 Web 组态功能测试用例表

输入/动作	期望结果	实际结果
从图元库选择图元控件拖拽至编辑区	在编辑区出现初始化图元空间并可自由拖动	图元可成功拖动
点击控件属性	可编辑控件属性值	可成功设置控件属性值
绑定数据至控件	相应控件可成功绑定数据并渲染控件	与期望吻合
更改页面背景	可成功更改页面背景	与期望吻合
三相电压、电流、功率数据发生变化	界面成功展示设备最新数据状态	与期望吻合，服务器接收到数据状态变化通过 WebSocket 实时推送到 Web 组态界面
电力变压器温度过高	警示灯从绿色变为红色	与期望吻合，警示灯变为红色

组态编辑界面如图 4.14 所示。左侧栏包括通用图元、“我的图形”、变电站专用基础图元模块，顶栏包括组态保存、发布、编辑、查看功能。“我的图形”模块可以保存基础图元构建成的复合图形，方便下次配电图组态直接拖拽生成，“我的图形”界面如图 4.15 所示。



图 4.14 组态编辑界面效果图

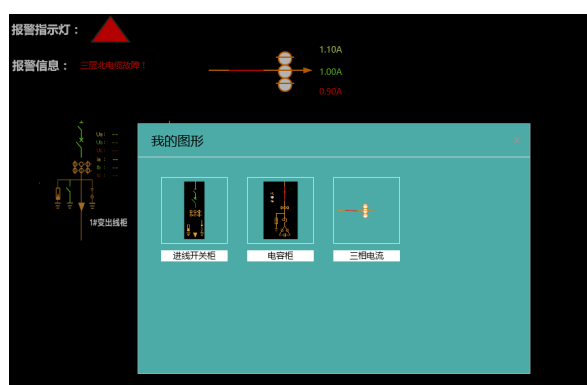


图 4.15 “我的图形”界面

点击组态模块顶栏“组态查看”按钮，可立即跳转至已组态的配电图界面显示，如图 4.16 所示，通过配电图的 Web 组态映射，达到可视化远程监测变电站实际电气接线情况。

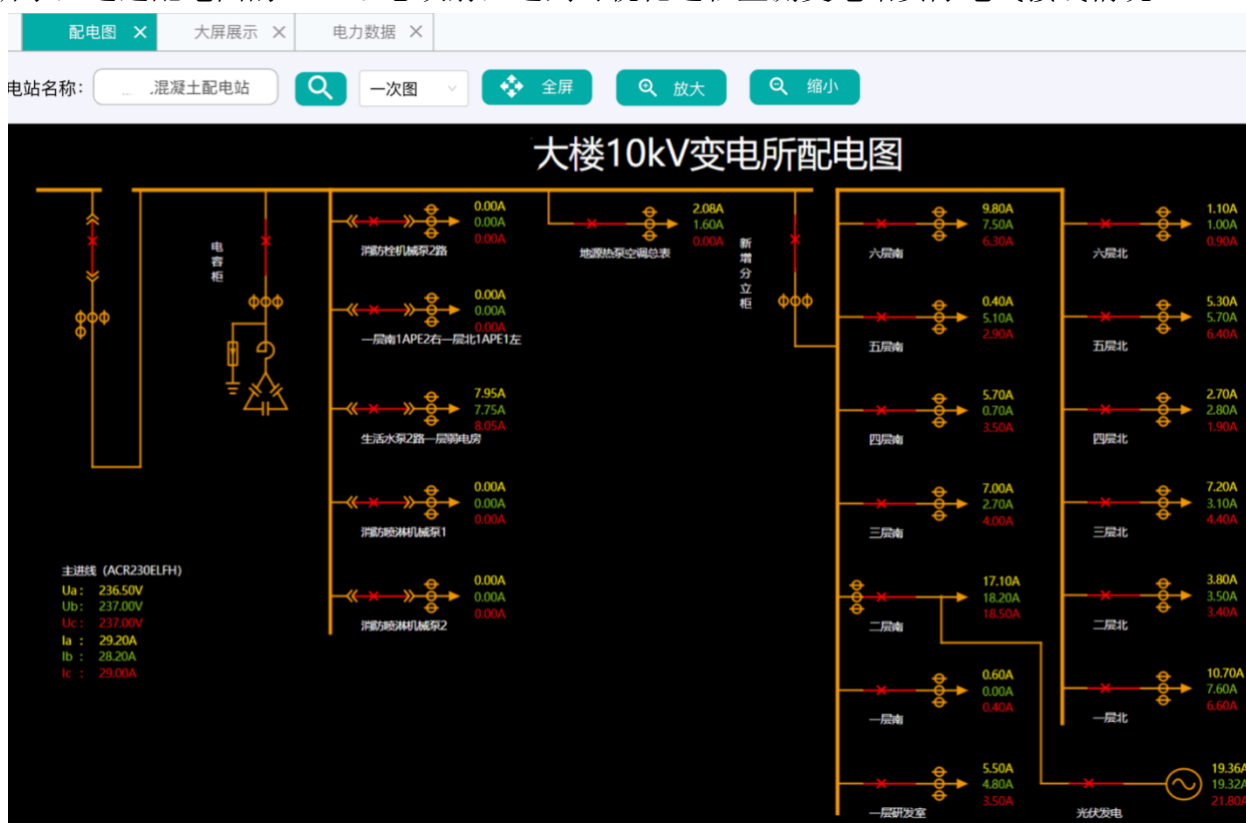


图 4.16 组态查看界面

测试表明，根据本文方案设计的 Web 组态各项功能正常，满足用户实际需求。

4.3.2 兼容性测试

组态界面对于兼容性要求很高，前文通过 Opera 浏览器进行了基本功能测试，下面分别使

用市面上求他四种主流浏览器对 Web 组态界面的兼容性进行测试，测试结果如图 4.17、图 4.18、图 4.19、图 4.20 所示，四种浏览器界面效果运行一致，组态界面元素呈现状态基本相同，功能正常，兼容性较好。



图 4.17 Chrome



图 4.18 Mozilla Firefox



图 4.19 Safari



图 4.20 Internet Explorer

4.4 本章小结

本章首先对目前组态软件进行研究分析，并指出如下缺点：构建发布复杂、复杂图形多输入量多输出量逻辑运算处理难、设备状态数据难以实时同步到客户端界面等。提出一种基于 SVG 与 WebSocket 的 Web 组态设计方法。利用 SVG 的 DOM 可操作性实现图形组态模块设计，并结合有向图处理复杂图形的输入输出量逻辑关系，使用 WebSocket 通信技术将设备运行数据状态实时推送至 Web 界面。

本章最后，进行相关测试，测试表明，本文设计的 Web 组态设计方法能简单快捷的构建 Web 组态界面，并能有效提高传感数据与 Web 组态界面同步的及时性。

第五章 总结与展望

5.1 总结

针对目前变电站监测系统的不足之处，本文基于物联网技术设计了一套智能在线监测系统。通过分析技术需求，设计智能系统整体框架：智能网关设计与变电站 PLC 控制柜设备物理连接，实时获取设备生产数据；利用 4G 通信技术与设备云平台保持长连接，将数据传至云端；开发 Web 应用程序，实现多个高可用的功能模块。

以变电站在线监测系统的总体设计为基础，结合客户端、云平台、设备的整体架构及业务需求，对监测系统的应用架构、故障预测方案和 Web 组态设计方法等进行深入研究，为变电站在线监测系统的搭建起到支撑作用。具体的关键技术包括卡尔曼滤波算法的改进优化、基于改进卡尔曼滤波算法的故障预测、基于 SVG 与 WebSocket 的 Web 组态设计方法。

本文立足于市场分析以及实际项目需求，实现了变电站在线监测系统，其中的主要工作和技术实现包括：

(1) 分析目前变电站在线监测系统技术上的缺点，结合 RS485、Modbus、4G 传输技术，将变电站设备入网，采集设备生产数据，并将数据存储在云端，实现变电站的远程监测。设计 Web 应用程序，并提供多项实用功能模块，保证监测系统的高可用性。

(2) 针对变电站异常状况不可预估或预测准确度不够的问题，提出一种基于改进 MGMO 的卡尔曼滤波算法故障预测方案，提高了变电站系统运行状态预测精度，减少了 60% 的预估精度的均方根误差，为故障的准确预测奠定了基础。

(3) 通过一种非线性收敛因子、记忆位置更新方程双策略来优化改进灰狼优化算法，测试表明改进灰狼优化算法相比标准灰狼优化算法性能更优。

(4) 针对传统卡尔曼滤波算法的预估精度的不足，通过 MGWO 算法迭代求出系统最佳初始噪声矩阵 Q 、 R ，从而有效降低卡尔曼滤波算法的发散性。

(5) 分析目前组态软件的不足，提出一种基于 SVG 与 WebSocket 的 Web 组态设计方法，利用 SVG 的 DOM 可操作性实现图形组态模块设计，提高了跨平台兼容性，并结合有向图理论简化组合图形多输入多输出的逻辑运算复杂度；使用 WebSocket 技术将传感数据实时推送至 Web 界面，有效降低了数据通信耗时。

5.2 展望

变电站在线监测系统在功能完整性、应用需求上达到了最初的设计目标，实现了变电站监测系统的智能化升级。但是由于个人经验不足、开发周期较短、变电站运行系统较复杂，

本系统仍然存在不足之处，需要进一步的改进：

(1) 当前在线监测系统是只适用于变电站设备的技术设计，扩展性还不够高，无法将其快速搭建到其他相关系统监测上。未来可以对智能网关接口扩展、云平台微服务架构升级，将监测系统打造为全工控行业的通用监测平台。

(2) 个人对卡尔曼滤波算法的使用经验还不足，只是验证卡尔曼滤波算法改进前后的性能差异，缺少和其他故障预测算法对比。

(3) 由于未来电网规模的持续扩大、设备接入数量的增加、数据量的激增，所以服务器的数据推送能力、负载能力以及浏览器的页面渲染性能等很多方面需要提高。

致 谢

时间如梭，不觉间研究生生涯即将结束，在杭州电子科技大学电子信息学院的每一次学习和每天的生活都是实实在在的经历，给了我最深刻的记忆。在论文完成之际，我要衷心的感谢在学习中帮助过我的每一位老师、同学和朋友，是他们的帮助和支持使我不断成长，谨此致以最深的感谢！

首先，感谢我的导师陈科明副教授，在论文的选题、构思、撰写到最终的定稿，陈老师都给了我悉心的指导和热情的帮助。在三年的学习生涯中，陈老师为我们提供良好的科研环境，让我学到了知识，也获得了实践锻炼的机会。在此谨向我最敬爱的陈老师表达我最诚挚的敬意和衷心的感谢。

感谢我的同门吴新虎、王彪、司端杨、郑杭明、安建伟、檀宇、俞峰、王奕怡同学，他们是我学习、工作和生活上的伙伴，也是我面对困难和挑战时的战友；感谢我的师兄师姐们在学习和生活上给我提供无私的帮助，他们是张凌云、周涛、周雪梅、蔡孟君、薛佳帆等；也感谢师弟师妹，为我提供的帮助。

感谢我的父母、弟弟及家人多年来对我的关心、帮助和理解，是他们为我营造了一个良好的求学氛围，没有家人的支持我难以完成学业。

感谢论文评阅和答辩委员会的老师，感谢你们的耐心，我一定会悉心听取你们宝贵的意见并积极改正。

最后，衷心地祝愿所有人，身体健康，工作顺利，让我们一起迎接更美好的明天！

参考文献

- [1] 中华人民共和国统计局.中国统计年鉴[M].北京: 中国统计出版社,2020.
- [2] 陶可鹏,倪惠浩,何雄辉. 变电站高压室断路器智能化改造研究与应用[J]. 机电信息, 2020, 000(006):51-52.
- [3] 李娜. 内蒙古地区智能变电站通讯技术与管理[D].天津大学,2018.
- [4] 张强. 基于组件化电力监测管理系统研究与实现.[D].哈尔滨工业大学, 2017:26-14.
- [5] 王丽娟. 基于 GPRS 的电力远程抄表系统数据分析监控软件设计.[D].西安理工大学, 2019:13-18.
- [6] 高阳. 基于 Web 的电力设备远程监控系统的设计与实现[D].电子科技大学, 2013.
- [7] Kannan N , Thomas S , Mobarak Y , et al. Enhanced data communication models for real-time power system monitoring in a distributed platform[J]. Transactions on Emerging Telecommunications Technologies, 2020(4).
- [8] Li X , Mba D , Yang Y , et al. Just-in-time Learning Based Probabilistic Gradient Boosting Tree for Valve Failure Prognostics[J]. Mechanical Systems and Signal Processing, 2020, 150.
- [9] 陈炜昕. 基于故障预测的船舶电力系统稳定控制方法研究[D].江苏科技大学, 2018.
- [10]瞿传柱. 改进支持向量机算法研究及其在 PHM 技术中的应用[D].哈尔滨工程大学,2016.
- [11]程声烽,程小华,杨露. 基于改进粒子群算法的小波神经网络在变压器故障诊断中的应用[J].电力系统保护与控制,2014,42(19):37-42.
- [12]傅建平. 基于无迹粒子滤波的锂离子电池剩余循环寿命预测研究[D].湖南大学,2016:12-13.
- [13]The Use of Configuration Concepti on in Software Development. Yang Y,Li M,Huang Y. Pacific-Asia Workshop on Computational Intelligence and Industrial Application IEEE . 2008Costa C, Scardi M, Vitalini V, etal. A dual camera system for counting and sizing Northern Bluefin Tuna (Thunnus thynnus; Linnaeus, 1758) stock, during transfer to aquaculture cages, with a semiautomatic Artificial Neural Network tool [J]. Aquaculture, 2009, 291: 161-167.
- [14]Sheng Q Z , Benatallah B , Maamar Z , et al. Configurable Composition and Adaptive Provisioning of Web Services[J]. IEEE Transactions on Services Computing, 2009, 2(1):34-49.
- [15]张洋. 基于 PLC 与组态软件的智能物料安装系统的设计与实现[D].南京理工大学,2015:43-08.
- [16]程功. 基于 Web 的煤矿监控组态软件的设计与实现[D].中国矿业大学,2019:23-67.
- [17]田娜. 基于组态软件技术的监控系统的设计与实现[D].浙江理工大学,2015:12-43.
- [18]Oussama Rholam,Mohamed Tabaa,Fabrice Monteiro et Abbas Dandache. Sm-art Device for Multi-band Industrial IoT Communications[J]. Procedia Computer Science,2019,155:660-665.
- [19]刘辉军,刘培锋,邱钰锋等. 基于开源框架及容器技术的微服务架构研究[J]. 电力信息与通信技术, 2018, v.16; No.178(06):90-94.
- [20]刘梓良. 面向 Java Web 的 3A 安全框架研究与设计[D].西安:西安电子科技大学,2017:45-52.
- [21]程春蕊, 刘万军. 高内聚低耦合软件架构的构建[J].计算机系统应用,2009,18(07) 9-22.
- [22]李华忠. 电缆接头温度监测技术的研究[D].哈尔滨理工大学.

- [23]Chen K , Yu J . Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach[J]. Applied Energy, 2014, 113(jan.):690-705.
- [24]黄文群.数字接收机中信号检测及自适应门限技术研究[D].天津大学,2011.
- [25]张文新. 基于神经网络算法的电力变压器温升推进预测[D]. 福建工程学院,2019.
- [26]文莹,肖明清,胡雷刚,方甲永.基于粗糙神经网络的航空电子设备故障预测研究[J].计算机测量与控制,2010,18(04):807-809.
- [27]郑含博,王伟,李晓纲,王立楠,李予全,韩金华.基于多分类最小二乘支持向量机和改进粒子群优化算法的电力变压器故障诊断方法[J].高电压技术,2014,40(11):3424-3429.
- [28]李川江.锂离子电池健康智能评估方法研究[D]. 南京航空航天大学,2015.
- [29]刘涛,解永春.一种自适应确定性采样滤波方法[J].信息与控制,2010(6).
- [30]Mirjalili S, Mirjalili S M, Lewis A. Grey Wolf Optimizer[J].Advances in Engineering Software,2014,69(3):46-61 DOI:10.1016/j.advengsoft.2013.12.007.
- [31]KoMaki G M,Kayvanfar V.Grey Wolf Optimizer Algorithm for the Two-stage Assembly Flow Shop Scheduling Problem with Release Time[J].Journal of Computational Science,2015,8:109-120.DOI:10.1016/j.jocs.2015.03.011.
- [32]徐强. 基于无线感知技术的有害气体源定位研究[D].上海应用技术大学,2018.
- [33]刘佳鸣, 况立群, 尹洪红,等. 灰狼优化的 k 均值聚类算法[J].中国科技论文, 2019.
- [34]段亚青,王华倩,乔学工.基于测距和灰狼优化的无线传感器网络定位算法[J].传感技术学报,2018,31(12):1894-1899.
- [35]魏政磊,赵辉,李牧东等.控制参数值非线性调整策略的灰狼优化算[J].空军工程大学学报(自然科学版),2016,17(3) :68-72.
- [36]McGarry, J. What if Minkowski had been ageusic? An alternative angle on diabetes[J]. ence, 1992, 258(5083):766-770.
- [37]Punit Rathore,James C.Bezdek,Sarah M.Erfani.Ensemble Fuzzy Clustering Using Cumulative Aggregation on Random Projections[J].IEEE Transactions on Fuzzy Systems,2018,26(3):1510-1524.
- [38]Porrati G M . 4D gravity on a brane in 5D Minkowski space[J]. Physics Letters B, 2000.
- [39]Rashida Adeeb Jhanum, Muhammad Asif Jan. Two new improved variants of gray wolf optimizer for unconstrained optimization. 2019,10.1109.
- [40]孟荣华. 金属结构件制造车间协同调度优化研究与应用[D].华中科技大学,2019.
- [41]Barnston A G . Correspondence among the Correlation, RMSE, and Heidke Forecast Verification Measures; Refinement of the Heidke Score[J]. Weather and Forecasting, 1992, 7(4):699-709.
- [42]姜丽丽. 工控组态软件(INTOUCH)研究与二次开发[D]. 河北工业大学.
- [43]冷波.基于 B/S Web 架构组态监控平台的应用[J].自动化应用,2019(09):72-74.
- [44]张文硕. 基于 Ajax 和 SVG 的监控系统人机界面设计与实现[D].华北电力大学,2018.
- [45]王彦文,赵永梅,张嘉璇.基于组态软件的配电网监控系统的研究[J].煤矿机电,2014(06):67-69+72.
- [46]张华. 担水沟煤矿安全监控系统的设计与实现[D].大连理工大学,2016.
- [47]郑丕珍. LBS 在旅游服务平台中的应用研究和示范[D].江西师范大学,2018.
- [48]赵程佳. 基于 SVG 的图形化业务监控系统设计与实现[D].复旦大学,2011.

- [49]李勇. 基于广义模块的产品变型设计方法及应用研究[D].武汉大学,2009.
- [50]刘钟凌. 顶点加权图的最密集子图算法设计与实现[D].广州大学,2018.
- [51]刘军. 基于 ZigBee 的智能停车场管理系统的设计与实现[D].哈尔滨工程大学,2012.
- [52]魏振超. 基于 3S 技术的特种车辆监控导航系统[D].重庆大学,2003.
- [53]Kaazing Corporation; Patent Issued for Enterprise Client-Server System And Methods Of Providing Web Application Support Through Distributed Emulation Of Websocket Communications (USPTO 10,686,850)[J]. Journal of Engineering,2020.
- [54]Amr S. Abdelfattah,Tamer Abdelkader,EI-Sayed M. EI-Horbaty. RAMWS: Reliable approach using middleware and WebSockets in mobile cloud computing[J]. Ain Shams Engineering Journal,2020.

附录

作者在读期间发表的学术论文及参加的科研项目

参加的科研项目：

- [1] 开关柜远程运维管理平台，横向课题
- [2] 智能电动自行车充电平台开发，横向课题

授权的专利：

- [1]***.一种基于改进灰狼优化算法的 5G 通信系统资源分配方法[发明]. 中国专利, 公开号:CN111787543A, 2020-10-16.