

学校代码： 10246
学 号： 15210720124

復旦大學

硕 士 学 位 论 文

(学术学位)

大型预制构件预应力采集管理系统研究

Research on Prestress Acquisition and Management System of Large
Pre-fabricated Components

院 系： 信息科学与工程学院

专 业： 通信与信息系统

姓 名： 王志文

指 导 教 师： 钱松荣 教授

完 成 日 期： 2018 年 3 月 30 日

指导小组成员名单

钱松荣 教授

目 录

| | |
|--------------------------------|----|
| 摘 要 | 1 |
| Abstract | 2 |
| 第一章 绪论 | 3 |
| 1.1 引言 | 3 |
| 1.2 研究前景及意义 | 3 |
| 1.3 论文内容、创新点以及组织结构 | 5 |
| 1.3.1 论文内容 | 5 |
| 1.3.2 项目创新点 | 6 |
| 1.3.3 论文的行文安排 | 7 |
| 第二章 无线无源传感器介绍及应力转换公式推导 | 8 |
| 2.1 无线无源传感器简介 | 8 |
| 2.1.1 传感器基本介绍 | 8 |
| 2.1.2 基于 RFID 技术的无线无源传感器 | 8 |
| 2.2 预应力转换与计算 | 9 |
| 2.2.1 预应力钢绞线 | 9 |
| 2.2.2 应变片与应变电路 | 10 |
| 2.2.3 应力转换装置 | 12 |
| 2.3 本章小结 | 15 |
| 第三章 系统架构总体介绍 | 16 |
| 3.1 系统需求分析 | 16 |
| 3.2 系统整体架构 | 17 |
| 3.3 系统安全性保证 | 19 |
| 3.4 本章小结 | 20 |
| 第四章 大型预制构件预应力采集管理系统 | 21 |
| 4.1 App 端 | 21 |
| 4.1.1 开发环境 | 21 |
| 4.1.2 工作流程 | 21 |
| 4.1.3 界面、接口与插件 | 22 |
| 4.2 Web 端 | 27 |
| 4.2.1 开发环境 | 28 |

| | |
|------------------------------------|----|
| 4.2.2 功能需求 | 28 |
| 4.2.3 Bootstrap 的引入与应用 | 29 |
| 4.3 服务器端 | 35 |
| 4.3.1 开发环境 | 35 |
| 4.3.2 工作流程 | 35 |
| 4.3.3 服务器配置 | 36 |
| 4.3.4 MVC 编程 | 37 |
| 4.4 数据库 | 39 |
| 4.5 本章小结 | 41 |
| 第五章 基于改进型 GM (1,1) 模型的预应力预测算法..... | 42 |
| 5.1 GM (1,1) 模型简介 | 42 |
| 5.2 GM (1,1) 模型的优化与改进 | 44 |
| 第六章 总结与展望..... | 47 |
| 6.1 工作总结 | 47 |
| 6.2 研究展望 | 47 |
| 参考文献..... | 49 |
| 攻读学位期间发表的学术论文..... | 52 |
| 致 谢 | 53 |

摘 要

伴随着全面建成小康社会的计划以及“中国梦”目标的逐渐实现,我国的各类基础建设也在更加紧锣密鼓的投入建设。其中,大型预制构件是基础建设中必不可少的一部分,大型预制构件的健康检测是十分重要的因为它关系着人民的生命财产安全。使用预应力钢绞线制作成的混凝土预制构件能够有效的避免混凝土的提前开裂,在实际生产中已得到了广泛地应用。

但是到现在为止,并没有一个成熟的系统可以用来对大型预制构件中的钢绞线预应力实现长期有效的监控,这也是这么多基础建设必须解决的一个隐患。本文为了实现对大型建筑的预应力预制构件的生命健康周期的监督,设计了一套完整的预应力信息采集和管理系统。

该采集管理系统的主要涉及硬件和软件两大部分。硬件主要是指工作于 125KHz 的基于谐振耦合技术设计的无线无源传感器,用来部署在预应力钢绞线上采集预应力信息。无源所以不需频繁更换电池,无线所以部署简单没有限制,综合以上所有优点我们选择无线无源传感器作为信息采集设备。

软件又分为 3 个部分: App 端、Web 端和服务器端。

App 端采用 C#(WPF)编写而成。App 端作为上位机软件需要在移动终端 Surface 上操作, WPF 作为微软开发的语言在 Windows 系统上有很好的兼容性。App 端主要负责接收来自读写器的预应力信息,然后分析、校验、入库。

Web 端是整个系统核心的部分,需要实现除了信息采集以外的所有功能,如:人事管理、传感器管理、预应力展示与预警、企业管理等。本文采用现在十分流行的 Bootstrap 框架实现。

服务器端主要工作是应用服务器的搭建和数据库服务器的设计。应用服务器采用 Tomcat+SpringMVC 的经典搭配,负责响应来自 App 端或者 Web 端的不同请求并返回结果。数据库服务器本文中 choice MySQL,主要负责与应用服务器通信,为客户提供查询、更新、事务管理、索引等多种服务。

最后,本文中提出了一种新的灰度模型预测算法,实现了对预应力数据的提前预警,将有可能出现的危害降到最低。

关键词: 大型预制构件监测, 预应力预警, WPF, Bootstrap, SpringMVC, GM(1,1) 算法

中图分类号: TN91

Abstract

With the gradual realization of the goal of building a well-off society in an all-round way and the goal of "China's dream," all kinds of infrastructure in our country are also being built even more intensively. Among all the infrastructure, large prefabricated components are an indispensable part. And health inspection for large prefabricated components is very important for it is related to people's life and property safety. Prefabricated concrete made of prestressed steel strands that can prevent concrete from cracking effectively have been widely applied in practical production.

So far, however, there is no mature system that can be used for long-term and effective monitoring of prestress in prestressed steel strand in large prefabricated components so that it is a hidden danger needs to be solved. In order to monitor the life cycle of prestressed precast members of large buildings, a complete prestress information acquisition and management system is designed in this paper.

This system divided into two parts: the hardware and software. The hardware mainly refers to wireless and powerless sensor based on 125 KHz resonant coupling technology, which is deployed on the prestressed steel strand and used to collect the prestress information. Powerless means never need a battery change, wireless means deployment is simple and unlimited. Taking all the advantages mentioned above into consideration, we use powerless and wireless sensor as our information acquisition devices.

The software, at the same time, is divided into three parts: the App, the Web and the server.

The App is written in C# (WPF). As an upper computer software, this App needs to operate on surface—the mobile terminal, and WPF, as a language developed by Microsoft, has a good compatibility on the Windows system. The main work of the App is to receive the prestress from card reader, and then analyze, check and save into the database.

The Web is the core part of the whole system, because it needs realize all functions except information acquisition such as personnel management, sensor management, prestress information display and early warning, enterprise management and so on. We use Bootstrap, a very popular front-end framework, to build Web end.

The main work on the server side is the construction of the application server and the design of the database server.

The application server is built with a classic combination——Tomcat+SpringMVC, which is responsible for responding to different requests from the App or the Web and return responses. We choose MySQL as our database. Database is responsible for communicating with the application server, providing customers with a variety of services such as query, update, transaction management, index and so on.

At the end of the paper, a new grey model prediction algorithm is proposed to realize the early warning for prestress data and reduce possible hazards

Key words: monitor prefabricate parts, prestress early warning, WPF, Bootstrap, SpringMVC, grey model algorithm

CLC: TN91

第一章 绪论

1.1 引言

随着“两个百年”奋斗目标的逐步实现，随着“一带一路”计划的提出与施行，我国不仅在国内加强基础建设，还把如高铁、桥梁等大型基础建设走向了国外，让中国及周边国家的朋友共享改革开放的结果。在建设生产各类大型基础建筑时，我们都需要用到大量的大型预制构件，如各类预制梁、预制空心板、预制靠件等。

预应力预制构件就是指预制构件（混凝土构件）在生产时预先拉伸混凝土构件中的钢绞线或钢筋，这样钢绞线回缩就会对混凝土产生一个压应力^[1]。当预应力预制构件正式服役受压时，因为承载而产生拉应力时就会需要先抵消压应力，然后才会拉伸预制构件。这种方式有效的发挥了钢筋的优势并阻值了预制混凝土构件过度拉伸而开裂。

现在很多的预制构件在生产时都预加了预应力，利用回缩力防止构件提早开裂。但是对正在服役的预应力构件，现在并没有什么很好的办法可以做到对预应力长期有效的检测。现有的检测方法都是机械物理的方法，大部分是如文献[2]、[3]中提到的方法：将钢绞索锚定，然后在钢绞线的一端采用锚力计或者测力计测量钢绞线的预应力，或者利用“物体在压力的作用下，其自振频率会降低；在拉力的作用下，其自振频率会增大”的原理，用频率法测量^[4]。但是无论是物理法还是频率法，都只能测量钢绞线两端的预应力值，无法测量构件中央的预应力值。但是现实情况中，由于存在应力松弛、混凝土结构蠕变、摩擦等情况，以及钢绞线和预制构件是非对称结构的，钢绞线两端的预应力值和中间的并不一样。物联网和 RFID 技术的发展，为我们解决这个问题提供了一个新的思路：我们可以在钢绞线的不同位置部署不同的应力传感器，然后通过无线技术将应力信息以电磁波的形式发送。读写器接收到预应力信息再发送给上位机用数据的校验、存储等工作。

在物理硬件解决预应力长期采集的问题后，如果能有一套相对应的软件实现预应力数据的管理、展示，以及一个相应的算法实现预应力数据的预测预警，就可以做到预应力长期有效的监控。

物理采集、展示管理、预测分析，这一整套系统为目前大型预制构件生产厂商或监督管理部门提供了一个很好的模板，这也是中国这么一个工业建筑大国必然要采用的管理趋势。

1.2 研究前景及意义

随着科技水平的提高，国内外的建筑行业也在蓬勃发展，更多安全可靠的生产、施工技术被提出、完善。预应力技术是通过在钢筋施加拉应力，进而防止了大型预制构件在荷载以后的变形和开裂，可以有效提高工程的耐久性，延长工程寿命，优化大

型建筑的工程质量^[5]。

由于大型预制构件生产厂商属于传统的制造行业，即使在互联网和物联网技术如此成熟的今天，工业互联网的生态系统依然冷冷清清。“工业 4.0”的概念提出已有若干年，除了在“智能生产”领域确实有长足进步，都采用了信息化、自动化的方式^[6]，但是在“智能物流”、“智能管理”、“智能决策”等领域，依然没有什么进展，不过这必然是接下来的发展趋势。对于像大型预制构件生产商、铁路交通运输部门、建筑质量检测部门等，大型预制构件的健康数据是非常重要的数据。但是对已服役的预制构件，并没有一种自动化、智能化的方式可以实现对预应力长期有效的采集监控。如引言中所述，现有的检测方法大多是物理机械的，一旦锚定在一端的测力计因为日晒雨淋或者其他原因损坏了，就无法再使用了也无法更换了。而且现在的这些测量法，无论是油压法还是传感器测量法，都只能测量其一端的预应力；而频率法和磁通量法又存在精度不够的缺点。

本文中使用的无线无源传感器（即 RFID 电子标签）及相关的应力转换装置，在大型预制构件生产中直接置入，与构件的钢绞线的相贴合。由于安装在大型预制构件内部避免了日晒雨淋，并且由于是一个无源的硬件，避免了更换电池的麻烦，所以具有寿命长、不易损坏的优点。此外，传感器采用 125KHz 无线传输数据，所以可以安置在钢绞线的任一位置采集任意一点的预应力信息。有效的解决了以上几种测量方法的缺点。

Windows Presentation Foundation(WPF)是微软公司的一个图形子系统，用于呈现基于 Windows 的应用程序中的用户界面^[7]。WPF 是基于 .NET 框架上的新型编程框架，它真正做到了将 UI 界面设计和程序设计分离^[8]，使得不论是界面设计还是逻辑设计都更加独立清晰。这也是它和微软最经典的编程技术 WinForm 之间最主要的区别，WinForm 的软件界面设计和软件逻辑是完全耦合在一起的，在强调高内聚弱耦合今天，这个设计显然是很糟糕的。作为微软新一代的图形系统，基于 DirectX 渲染技术的 WPF 带来了当时前所未有的 3D 界面、3D 动画以及半透明窗体效果等，给了客户强大的视觉体验。MVVM 设计模式和今天很流行的 AngularJS 以及 React 是一致的，强调的是数据绑定的特点，当 Model 变化时，View-Model 会自动更新，View 也会自动变化，很好做到了数据的一致性。此外，它对动画、3D、高级图形高度支持，对复杂界面开发具有超高效率，同样是 C++ 需要半年开发出来的界面，用 WPF 可能只需要一两个月。这些优点使得 WPF 成为了很多大型企业定制企业 ERP 的优选。另外，众所周知的 QQ 概念版也是用 WPF 开发的。由于客户端需要涉及天气的呈现、预应力数据的采集和展示，涉及到很多图表，所以本文中选择 WPF 作为客户端开发技术。

Bootstrap 是现在“网红”级别的前端框架，它由 Twitter 的两个员工 Mark Otto 和 Jacob Thornton 合作开发而出，并与 2011 年在 GitHub 上开源发布。它是基于 HTML、

CSS 和 JavaScript 的灵活简单的 Web 开发框架。Bootstrap 之所以这么火，最主要的原因就是因为它是响应式的框架，所谓响应式就是指无论你是在什么分辨率的电脑上，无论你是在 PC、Pad 还是不同手机上，它都能自适应的调整网页布局^[9]，以最优化的方式呈现给访问者。此外，Bootstrap 还具有成熟、拓展性强、拥有现成 UI 组件、有完善的文档、开发速度快^[10]等优点。Bootstrap 自开源以后，很多公司的网站都利用它进行开发，比如：星巴克官网，Ghost 中文网，Jekyll 中文网等。考虑到它技术成熟，插件、UI 组件丰富，开放快上手效率高等优点，我们也选择 Bootstrap 作为我们网站的主框架。

SpringMVC 是 Spring 框架的一个重要组成部分^[11]。Spring 是现在最受欢迎的企业级 Java Web 开发框架之一，几乎国内所有的大型互联网公司你都能看到它的影子。Spring 是一个开源的轻量级的 Java 平台，由 Rod Johnson 开发，并与 2003 年 6 月根据 Apache 许可首次发布。Spring 基本版本只有 2M 大小，在进行 Web 开发是支持其他组件的拓展的。SpringMVC 实现了 Model-View-Controller(MVC)的概念，MVC 模式有助于了应用程序的不同逻辑如输入逻辑、业务逻辑和 UI 逻辑的分离，为这些元素提供了松散耦合的良好环境，这样的架构可以帮助开发人员搭建结构清晰、可复用性好、维护方便的 Web 应用^[12]。

至此，我们整个系统的结构已经很清晰了，这是一个同时基于 C/S 和 B/S 混合结构的大型预制构件预应力采集管理系统：硬件部分由无线无源传感器和相应的应力转换装置构成，主要任务是预应力信息的采集；client 端由 C#(WPF)编写而成，功能为串口数据的读取，预应力数据的校验、入库，以及查询展示；browser 端作为系统企业级入口，具有除了预应力读取、录入以外的所有的功能模块；server 端负责响应来自 client 端和 browser 端的所有请求并与数据库交互。

该系统可以作为现代各类物联网相关企业的数据管理系统，为如大型预制构件预应力检测之类的至今未实现自动化智能化的工程应用提供了“物联网+互联网”的解决方案。客户端 App 作为物联网端的上位机系统，负责物联网内的数据采集校验等工作。浏览器端需要实现企业级信息管理系统的所有功能：实现数据的处理、展示和预警等数据功能，以及个人管理、组织管理、等各种系统功能。该系统为所有物联网数据的长期采集和监测提供了一个模板系统，具有广阔的应用前景。

1.3 论文内容、创新点以及组织结构

1.3.1 论文内容

在实验室前辈研究的无线无源传感器的基础上，结合现在大型预制构件预应力检测存在的如检测精度不够、探测器或传感器易损坏，无法长期实施监控等诸多弊端，创新性的将装有全桥应变片的无线无源传感器作为预应力采集的物理设备嵌入大型预

制构件之中，有效避免了各种人为或自然因素导致的损坏，实现了长期有效的监控。此外，我们利用 WPF、Bootstrap、以及 SpringMVC 等前后端相关技术，对应现场采集和后台人员、数据管理，基于 C/S 和 B/S 混合架构设计了一套采集管理管理系统，并改进了传统 GM (1,1) 模型算法，是的对预应力走势的预测更加准确。

本文的主要内容如下：

(1) 在实验室前辈设计的基于 125KHz 的无线无源通信模块基础上，根据工程力学和电路知识，将预应力转换装置中的力与电信号转换关系做了精确的数学推导，对无线无源传感器对大型预制构件中钢绞线的预应力长期监控实现了理论基础，以及为日后正式生产中的部署提供了一种解决方案。

(2) 针对现在对各类大型预制构件中钢绞线的预应力检测方法还停留在机械人工方法的现状，更有甚者无法长期监控、无法监控的缺点，在内容(1)的基础上，设计一个 C/S 结构的信息采集、校验、查询软件。Client 端由 C#语言编写，Server 端与 B/S 架构的服务端一致，由 Java 编写。工作时 RFID 读写器读取无线无源传感器的采集的预应力数据，然后上位机软件串口读取 RFID 读写器中的数据，解析、校验以后与服务器交互。C/S 结构实现的员工登陆、信息采集、天气查询，预应力查询等一系列查询功能。

(3) 为了长期有效监测预制构件的健康状况，生命周期以及预应力大小等数据及其相关规律，设计了一套 B/S 架构的企业系统。在 B/S 架构中，前端使用 Bootstrap 编写，后端使用 Java 的 SpringMVC 框架。该系统也可以拓展为企业的信息管理系统，传感器、预应力的查询展示只是作为数据展示功能的一部分，还实现了如个人管理、邮件管理等个人事务功能以及人事管理、部门管理、以及通知公告等系统功能。

(4) 采集储存预应力数据的最终目的是从已有的数据出发，发现预应力的变化规律，从而预测应力的未来走势，做到提前预警。灰度模型 (GM (1,1)) 是现在工程和经济等领域经常用到的一种预测模型算法，所谓灰度模型即有众多不确定的因素，无法从物理模型本身去建立一个合适的公式模型，所以从数据本身规律出发去拟合计算。本文中对已有的传统 GM (1,1) 算法进行了改进和优化，考虑到不同情况下背景值的计算有很大出入，根据实时情况我们用加权平均改变了背景值的计算公式。经数据检验，改进型 GM (1,1) 模型算法比传统算法准确度更高。

1.3.2 项目创新点

本文（项目）的创新点有以下三项：

(1) 对于传统的贴片式应变片无法直接贴在钢绞线上的问题，提出了一个预应力装换装置，并数学上严格证明了力与电数学的关系——即预应力计算公式，证明了其可行性。

(2) 对传统的 GM (1,1) 模型算法进行了优化改进。经数据检验，其预测准确度

更高, 相比传统 GM (1,1) 模型, 改进型模型平均误差降低了 20%左右。

(3) 设计了一套基于 C/S 和 B/S 混合模型的大型预制构件预应力检测管理系统。该系统支持离线采集处理数据, 实现了预应力数据的采集、展示、管理以及预测分析。在国内, 成熟稳定的预应力检测管理依然比较少。

1.3.3 论文的行文安排

本文共分为七个章节, 内容概述如下:

第一章为绪论, 阐明本文的研究背景、国内外的研究现状以及本文的前景及意义, 简单介绍了本项目中用到的各种技术, 并说明了文章的内容结构安排。

第二章为无线无源传感器的简单介绍以及对无线无源传感器应力转换做了严格的数学推导。

第三章为系统总体架构的设计和介绍, 介绍了 B/S 和 C/S 混合结构的工作流程, 分析了 client 端、browser 端、sever 端分别要实现的功能。

第四章为系统的具体分析。主要内容包括 App 端、前端、服务器端、数据库服务器的具体与部分核心代码的分析。为 App 端功能介绍, 包括工作流程图, 相关界面设计以及使用的接口和插件。Web 端分析了作为一个企业网站的需求和功能, 分析了前端设计、UI 设计以及网站页面布局等。服务器(server 端)的设计, 介绍了 WEB 服务器的配置, 工作流程, 如何与数据库服务器交互以及核心代码的解析。数据库端给为了设计数据表的要求和规范。

第五章介绍了传统 GM (1,1) 算法模型, 并分析了该模型算法的缺点与不足, 并提出了一种基于因子加权思想优化背景值序列的改进型 GM (1,1) 算法, 用真实数据对预应力的预测分析进行检验, 证明该算法的预测更加准确。

第六章对全文进行了总结, 并结合预应力采集管理系统的实际开发和运行情况, 进行了展望分析。

第二章 无线无源传感器介绍及应力转换公式推导

2.1 无线无源传感器简介

2.1.1 传感器基本介绍

传感器作为一种检测装置，作为物联网必不可少的组件之一，可视为人类感官的延伸，可以把我们想要的温度、湿度、压力等信息转换为电信号的形式传输、记录、储存。现在最广泛的应用的传感器是无线传感器，因为无线传感器不需要提前布线，可以节省前期施工成本，并且部署起来也更加灵活^[13]。但是随着无线传感器地广泛应用，它的各种缺点也暴露出来了，最突出的问题就是电源供电^[14]。因为绝大部分传感器，由于部署环境恶劣或更换电池不易等原因，在部署以后都是不再回收的。一旦传感器中的能量消耗完了，传感器也就死亡了。所以如果能解决传感器的持久供电问题，那么传感器也就能加速其大规模推广的步伐^[15]。

为解决此问题，无源传感器应运而生。顾名思义，不需要给无源传感器提供能量（电池），它就可以正常工作。那么它工作时能量从何而来？无源传感器的分很多种，有声表面波无源传感器^[16]，有光电式无源传感器^[17]，有压电式无源传感器^[18]，以及还有根据 RFID 技术制成的无线无源传感器等。这些无线无源传感器，既保留了普通无线传感器部署简单、自适应等优点，又具有可持续工作、无需更换电源的优点。

2.1.2 基于 RFID 技术的无线无源传感器

射频识别（RFID）是一种利用电磁耦合技术实现特定目标识别并读写数据的无线通信技术。RFID 技术是一种非常常见的技术，我们平时所熟知的身份证、公交卡、学生卡以及高速公路道路自动收费等都是利用 RFID 技术工作的。RFID 有五种频段的空中接口协议，分别是 125~134KHz, 13.56MHz, 2.45GHz, 860~960MHz 以及 433.92MHz。由于我们的无线无源传感器（既 RFID 芯片）需要放置在大型预制构件里面，几十厘米厚的混凝土层会对电磁波产生阻碍作用，所以对于频段的选择很重要。2.45G 和 960M 等高频信号虽然具有方向性好、通信距离远的优点，但是由于其穿透性弱，无法穿透厚厚的混凝土预制构件，所以不适合用于此处。在文献[19]中，实验室师兄选择 125KHz 的频率，通过优化带负载匹配电路的串并联混合谐振电路，极大的提升了耦合负载电路的通信距离。已证明，即使隔着厚厚的钢筋混凝土，也能在最大的范围 38cm 实现传感器和读写器的良好通信。RFID 读写器有传感器的通信、加解密的等过程在文献中也有具体介绍，此处不再赘述。论文在传感器和读写器良好通信的基础上开展研究。

2.2 预应力转换与计算

2.1.1 预应力钢绞线

预应力钢绞线是由很多根钢丝高强度钢丝绞合而成并经过并消除应力处理合成的高强度钢铁制品。钢绞线如图 2-1:



图 2-1 混凝土管桩生产管理系统功能架构图

我们要检测的预制构件的预应力其实就是钢绞线的预拉力。在相同大小的力的作用下，不同物体的形变大小只和该物体的弹性模量有关。如果是一般的线性形变，弹性模量即杨氏模量，定义为：对一根细杆施加一个拉力 F ，这个拉力除以杆的截面积 S ，称为“线应力”，杆的伸长量 dL 除以原长 L ，称为“线应变”。线应力除以线应变就等于杨氏模量，即：

$$E = \frac{\frac{F}{S}}{\frac{dL}{L}} = \frac{F \cdot L}{S \cdot dL} \quad (2-1)$$

工程上，不同的材料、型号的钢筋有不同的弹性模量，常用的钢筋及钢绞线弹性模量如下图所示：

| 钢筋的弹性模量 ($\times 10^5 N/mm^2$) | |
|--|--------|
| 牌号或种类 | 弹性模量 E |
| HPB300 钢筋 | 2.10 |
| HRB335、HRB400、HRB500 钢筋 HRBF335、HRBF400、HRBF500 钢筋 RRB400 钢筋 | 2.00 |
| 钢绞线 | 1.95 |

图 2-2 钢筋弹性模量表

由于工程上，一旦钢筋或钢绞线制作完成，它的弹性模量也就是一个定值，所以我们可以把E当作是一个已知的常数；同理，S和L也是已知常数，根据公式(2-1)我们可以得到：

$$F = \frac{E \cdot S}{L} dL \quad (2-2)$$

所以理论上我们只要测得钢绞线的形变大小，就可以计算出预应力的。但是在各类大型预制构件中，如高铁 T 型梁和工型梁，它们在服役时其中的钢绞线应变一般为 3.5%，但是传感器上的应变片应变极限为 2%。所以直接把压力应变片粘贴在钢绞线上将导致应变片无法工作在线性区而无法正确测量，甚至还有可能导致应变片的断裂。所以直接测量形变的方法是不可行的，我们需要一个装换装置间接测量钢绞线预应力的。该装置将在 2.1.3 节介绍。

2.1.2 应变片与应变电路

电阻应变片是指在压力、电、磁、光等作用下阻值会发生变化的半导体元器件。本文使用的压变效应即当应变材料在收到外界作用力时其电阻值也发生相应变化的现象。若一根圆形切面的金属丝电阻系数为 ρ ，长为 L，横截面积为 S，则电阻值计算公式为：

$$R = \rho \frac{L}{S} \quad (2-3)$$

当金属丝在轴向力作用下：长度延伸、面积缩小、电阻率变化，电阻的阻值也随之变化。电阻应变片可以把被测压力引起试件应变的变化转换成电参数 R 的变化^[20]。但是这是一种十分微小的变化，无法直接用欧姆表等设备直接测量，所以我们需要利用电桥电路将电阻变化转化为电流信号。直流电桥结构图如图 2-3 所示：

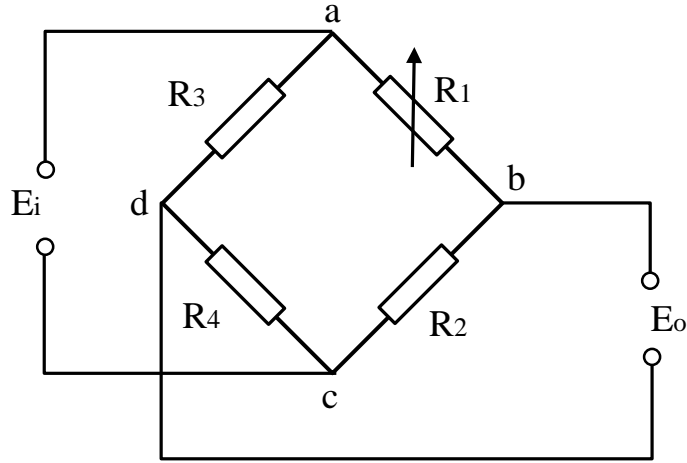


图 2-3 单臂电桥

其中 R_1 是应变片, R_2, R_3, R_4 是普通电阻, 当应变片不受力时, $R_1=R_2=R_3=R_4=R$ 。此时 b, d 两点的电位都是 $E_i/2$, 输出电压 E_o 为 0。当应变片在外力的作用下发生形变, 阻值变化为 ΔR , R_1 阻值变为 $R + \Delta R$, 此时 b 点电位发生变化, E_o 不再为 0。输出电压与电阻值变化的关系为:

$$E_o = E_i \left(\frac{R + \Delta R}{2R + \Delta R} - \frac{1}{2} \right) = E_i \left(\frac{\Delta R}{4R + 2\Delta R} \right) \approx \frac{1}{4} \left(\frac{\Delta R}{R} \right) E_i \quad (2-4)$$

衡量应变材料特性的一个重要参数是灵敏度, 他的定义如下:

$$K = \frac{\Delta R / R}{\varepsilon} \quad (2-5)$$

其中 $\varepsilon = dL/L$, 即应变, $\Delta R / R$ 为电阻的变化率。将公式(2-5)代入公式(2-4)可以得到:

$$E_o = \frac{E_i}{4} K \varepsilon \quad (2-6)$$

(2-6)中灵敏度 K 和弹性模量相似, 一旦物品成型, 就是一个固定的常数, 所以我们可以把它看作一个已知常数, 因此可以得到结论: 输出电压正比于应变。

但是大型预制构件一般都工作在室外有可能遇到各种天气和温度。温度和湿度的变化都有可能导致半导体材料形状或材料特性产生变化, 最终导致阻值的变化。所以在实际工程应用中, 我们还要考虑零点漂移和温度补偿^[21]等问题。

如图 2-4 所示, 当电桥电路的四个臂都变成了相同的应变片, 这个电路也变成了全桥电路。全桥应变片就是利用这个结构制成, 它的四个臂都是相同的普通应变片。图中的 R_a, R_b 和 R_t 是调零电路 $R_a=R_b=R_1=R_2$ 。当全桥应变片投入工作时, 虽然单个应变片阻值会受到温度的影响, 但是四个应变片同时发生相同的变化, 所以依然满足电桥平衡公式^[22]:

$$R'_1 R'_4 = R'_2 R'_3 \quad (2-7)$$

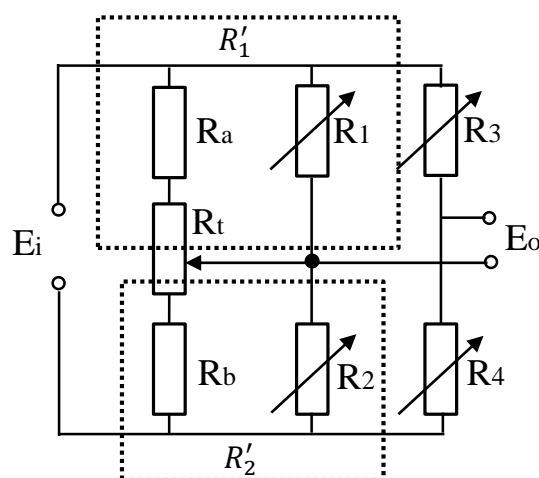


图 2-4 带调零电路的全桥电路

除了具有温度补偿的优点，全桥应变片的灵敏度是普通应变片的 4 倍。综合以上优点，我们选择全桥应变片而不是普通应变片测量大型预制构件钢绞线的预应力。

2.1.3 应力转换装置

2.1.1 节已经提到，用应变片直接测量钢绞线的应变会出现测量不精确或者应变片损坏的现象，所以我们需要设计一个应力转换装置来间接测量计算钢绞线预应力。如图 2-5：

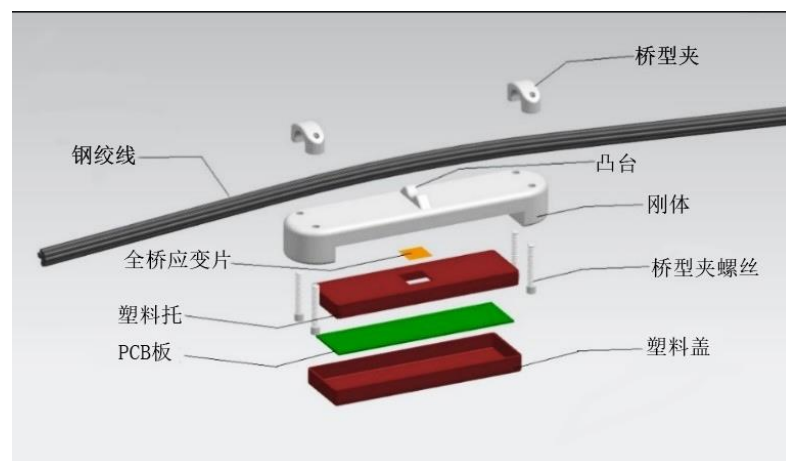


图 2-5 预应力转换装置

转换装置的工作原理是:将装置固定在钢绞线以后，由于桥型夹和凸台不在一条直线，所以钢绞线在施加预应力的情况下会对凸台产生压力。刚体正下方贴着应变片，所以当刚体在压力的作用下发生微小形变时，应变片也发生相应形变。PCB 板即无线无源传感器，上面印刷的电路会将电阻变化以电信号的形式通过电磁波发送出去。可以将钢绞线受力模型简化为图 2-6：

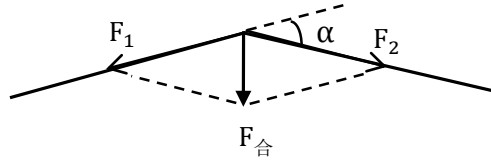


图 2-6 预应力与正压力分析模型

假设预应力钢绞线在凸台的压力作用下发生微小形变，形变角 α 趋向于 0。根据经典力学知识我们可以知道，当预应力为 F 时， $F_1 = F_2 = F$ ，合力为：

$$\begin{aligned}
 F_{\text{合}} &= \lim_{\alpha \rightarrow 0} 2 \cdot F \cos\left(\frac{180^\circ - \alpha}{2}\right) \\
 &= \lim_{\alpha \rightarrow 0} 2 \cdot F \cos\left(90^\circ - \frac{\alpha}{2}\right) \\
 &= \lim_{\alpha \rightarrow 0} 2 \cdot F \sin \frac{\alpha}{2} \\
 &= \lim_{\alpha \rightarrow 0} \alpha \cdot F
 \end{aligned} \tag{2-8}$$

根据公式(2-8)我们可知合力与角度成正比，由于装换装置是尺寸是固定的，形变角 α 即桥型夹和凸台形成的角度也是固定的，即可作为一个已知常量。

接下来分析转换装置的刚体的受力情况，将刚体受力简化为简支梁模型。如图 2-7：

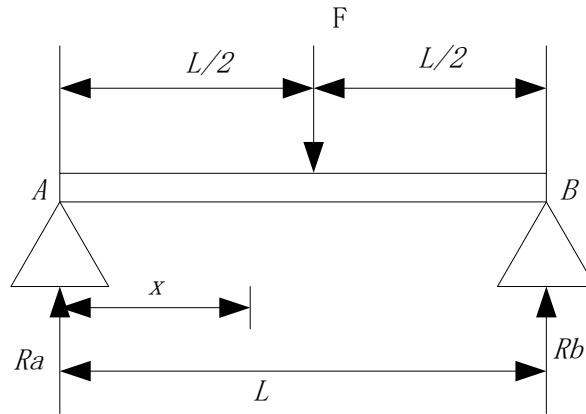


图 2-7 简支梁模型

图 2-7 中，简支梁中间收到力 F 作用，则 A、B 两点的反作用力 $R_a = R_b = F_{\text{合}} / 2$ 。弯矩是指弯曲构件所需要的力矩，在简支梁集中力模型下弯矩 M 与到端点 A 距离 x 的关系是：

$$M(x) = R_a x = \frac{F_{\text{合}}}{2} x \quad (0 < x < \frac{L}{2}) \tag{2-9}$$

$$M(x) = R_a(L-x) = \frac{F_{\text{合}}}{2}(L-x) \quad (\frac{L}{2} < x < L) \quad (2-10)$$

梁的横截面某处的线应力大小^[23]为:

$$\sigma_x = \frac{My}{I_z} \quad (2-11)$$

式中, x 是指梁的横向, y 是纵向, z 是宽度方向, 即垂直纸面方向。 M 是 x 处力矩, I_z 是截面相对 z 轴的轴惯性矩, 由于仅和几何尺寸相关所以可以看作常量。将公式(2-9)代入(2-11), 并假设刚体的厚度为 a , 则刚体下方应变片受到的线应力大小为:

$$\sigma_x = \frac{x F_{\text{合}} a}{2 I_z} \quad (2-12)$$

由于 I_z 和 a 都是常数, 公式(2-12)可以简化为:

$$\sigma_x = k_1 \cdot F_{\text{合}} \cdot x \quad (2-13)$$

而应力和应变的关系为:

$$\varepsilon_x = \frac{\sigma_x}{E} \quad (2-14)$$

E 即杨氏模量, ε_x 为刚体在 x 处的应变, 即:

$$\varepsilon_x = \frac{\Delta L_x}{L_x} \quad (2-15)$$

故总应变为:

$$\frac{\Delta L}{L} = 2 \int_{\frac{L}{2}-\frac{a}{2}}^{\frac{L}{2}+\frac{a}{2}} \varepsilon_x dx \quad (2-16)$$

将公式(2-8)、(2-13)代入, 得到:

$$\varepsilon = \frac{\Delta L}{L} = k_2 F \quad (2-17)$$

其中 k_2 为比例系数。根据(2-17)可以得到一个结论: 装换装置刚体的应变与钢绞线预应力成正比。

结合公式(2-6)和(2-17), 可以得到重要结论: 应变片的输出电压与预应力成正比。

$$F = K E_o \quad (2-18)$$

K 为比例系数, 由应变片灵敏度、弹性模量、输入电压以及凸台角度等决定, 大型预制构件生产时, 可以测得该参数的大小。

至此, 在理论上已证明无线无源传感器及相关装换装置可以实现钢绞线预应力的长期、有效检测。

2.3 本章小结

本章主要介绍了系统要用到的硬件知识，为系统的开发做好了硬件上的准备。前半部分为对传感器的基本介绍，选择无线无源传感器的原因以及基于 **RFID** 技术的无线无源传感器的应用举例。在第二小节介绍了预应力转换装置并从数学上严格推导了预应力大小与输出电信号的线性关系，为传感器采集应力数据证明了可行性和可实用性。

第三章 系统架构总体介绍

第二章为该系统的信息采集奠定了硬件的基础。本文研究的重点的设计一个大型预制构件预应力的管理系统采集。通过 WPF、Bootstrap 以及 SpringMVC 等框架，设计了一个基于 C/S 与 B/S 混合模式的系统，实现了对预应力信息的采集、管理、展示、预测预警，传感器信息的录入、管理以及员工信息的管理等功能。

3.1 系统需求分析

目前，包括铁路运输部门、预制构件生产厂商、质检部门在内的国家部门或企业都已经意识到对大型基础建筑的长期监测对人民群众生命财产的重要性，此外服役数据的变化情况可也作为一个正反馈让生产商调整工艺。然而现有的大型预制构件健康监测还是人工、费时费力的，都是采用员工现场读取预应力数据或检查混凝土构件是否开裂来判断构件是否健康。并且是以手工、纸质的方式记录，不利于资料数据的统一。为了解决无法监测、无法长期监测、低效的预应力检测等问题并做到智能分析、智能决策、提前预警，提取出几个具体需求：

(1)生产信息化管理

需要在生产大型预制构件过程中，直接将无线无源传感器置入构件中，传感器 ID 是唯一的，每一个传感器对应一个构件。加装固定好以后，再在后台系统录入相应的传感器信息、预应力初始值、构件编号、转换系数等参数。以后就能在构件服役时实现对预应力构件的长期管理和信息采集。

(2)智能采集

在构件正式服役以后，对预应力信息的采集是通过 RFID 读写器完成的。故需要设计一个 client 端的 App，在移动终端上实现与 RFID 读写器的串口通信，并检验数据的正确性，并实现与远程数据库的交互。此外，为了保证预应力数据的正确性、安全性以及责任到人，App 需要实现员工的登陆功能。

(3)智能分析和预警

在过去，是去现场查看混凝土构件是否开裂判断构件是否健康，但是这时为时已晚。为了解决这个问题同时也是采集大量预应力数据的目的，需要从已有的数据实现智能分析。在 browser 端需要实现包括同一构件随时间预应力变化情况以及不同构件总体预应力变化与负载的关系分析两方面。从时间和空间两个维度对大型预制构件健康已经预应力变化进行监测分析。并通过 GM (1,1) 算法预测，对于马上到那临界区的构件提前做出预警，以便提前整修加固。

3.2 系统整体架构

根据上述需求分析,并考虑到不同的工作环境与工作要求。设计了基于 C/S 和 B/S 混合结构的系统架构,如下图所示:

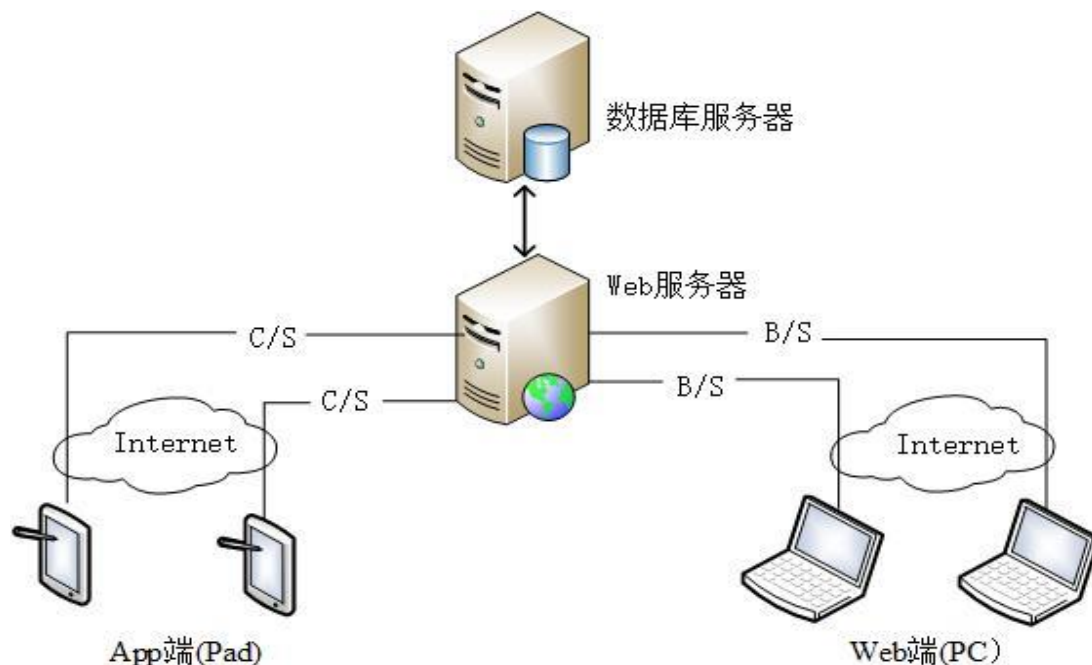


图 3-1 大型预制构件采集管理系统架构图

系统总共分为三大部分: App 端, Web 端和服务器。App 应用和服务器构成 C/S, PC 和服务器构成 B/S 结构。传统的 C/S 结构都是 client 端通过局域网直接与数据库服务器连接通信^[24], 不需要通过 Internet 与 Web 服务器相连。但是此处的 client 端却与服务器通信, 仿佛与 B/S 端并无区别, 似有多余之嫌。接下来会做具体解释。

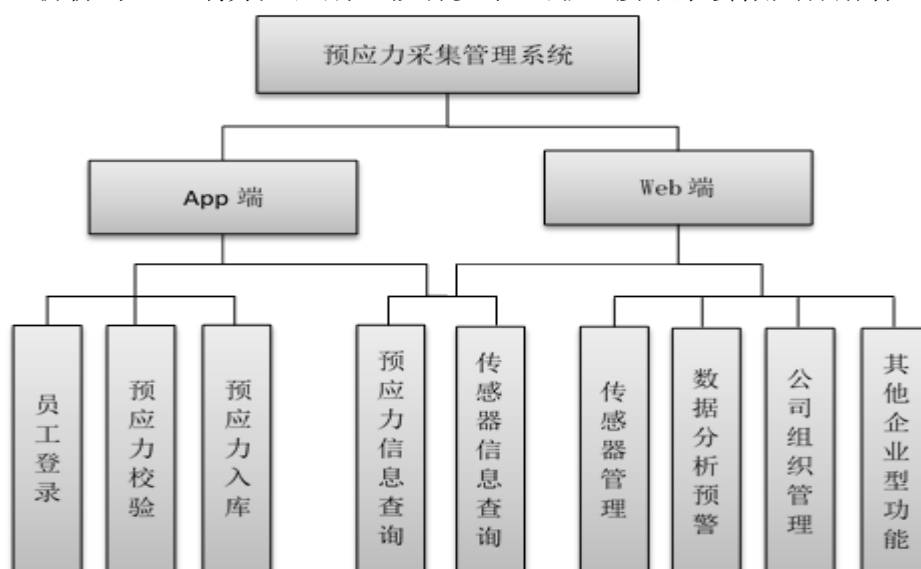


图 3-2 大型预制构件采集管理系统功能图

系统的功能图如图 3-2 所示，App 端和 Web 端要实现的功能不尽相同，App 端主要实现预应力的采集，而 Web 端需要实现除了信息采集之外的所有功能。

系统各部分功能介绍如下：

(1)大型预制构件信息采集管理系统 App 端

主要负责实现“采集管理”的“采集”，是数据的来源。当置入传感器的预制构件正式服役以后，就需要实现对预应力数据的监测。由于读取无线信号需要 RFID 读写器，所以 App 的信息采集实际是基于串口通信的。信息采集需要在机场、铁路、高架、隧道等各种室外环境完成。考虑到 App 的承载设备需要具有便携性并可以实现串口通信，最终我们选择 Windows 平板电脑作为移动终端。与此相应，选择 C#的 WPF 框架作为 App 的开发框架，WPF 是微软开发的运行与 .NET Framework 之上的开发框架，所设计的程序在 Windows 环境有很好的兼容性。如图 3-2，App 需要完成登陆、数据入库以及查询等功能，但是工作地点由于在室外各种地形，所以不可能像传统 C/S 架构一样建设局域网，而是通过 Internet 与 Web 服务器交互。

那么既然如此为什么不把所有功能都集成在 B/S 架构上呢？有两点原因，一是因为使用 C#开发串口通信更加简单方便，浏览器开发串口通信还需要下载支持的控件，不同的浏览器对控件的支持兼容程度也不同；第二点也是最重要的一点，是因为 B/S 架构能正常工作的基础是网络的畅通，B/S 架构所有的控制和计算都是有服务器完成，在无网络环境，浏览器是无法离线打开与运行的。但是如前面多说，移动终端工作的环境实在室外，有可能是隧道等信号不好的地方。这时 B/S 结构就无法工作了。但是 C/S 架构有一个好处：可以充分发挥 client 端的运算功能，即使没有网络也可以完成部分功能。所以这个大型预制构件信息采集管理系统 App 端具有离线登陆、离线存储等功能。可以将数据线储存在本地数据库，然后等恢复网络时再将数据通过网络上传至数据库

(2)大型预制构件信息采集管理系统 Web 端

实现除信息采集之外的所有系统功能。当信息采集完毕以后，需要实现需要有大量的人机交互的操作。由于展示、分析这些功能的实时性要求并不高，重点为用户交互的舒适性及展示的丰富性。B/S 结果作为现在主流的软件架构，开发维护简单，支持大量的控件和插件，对于需要实现的各类展示功能以及其他企业级功能都能轻易实现。Web 端使用现在比较火的 Bootstrap 框架搭建。Bootstrap 是一个响应式前端框架，在任何浏览器以及不同分辨率是终端都有良好的交互界面。

Web 端需要实现的功能有：传感器基本信息查询与修改；预应力的查询与展示；预测预警功能；邮件功能；以及通知公告等其他公司级企业功能。

(3)Web 服务器

Web 服务器就是网络上给浏览器提供信息浏览、下载等信息服务的计算机服务。

目前主流的三大 Web 服务器分别是：Apache、Nginx 和 IIS^[25]。其中 IIS 是微软公司开发主推的服务器，但是因为它只支持 Windows 环境，所以市场占有率不如 Apache。Apache 作为世界第一大 Web 服务器，市场占有率依然达到 60%，它几乎可以在目前所有的平台运行，如 Unix、Linux 以及 Windows。考虑到开源性、兼容性以及开发维护的简易性，我们选择 Apache 旗下的 Tomcat 作为服务器。并选择 SpringMVC 作为 Web 容器开发的框架。

Web 服务器在本系统中的作用就是对来自 App 或者浏览器的 HTTP 请求进行拦截过滤，与数据库服务器交互，并返回不同的结果。

(4)数据库服务器

数据库服务器即数据库。用于储存企业基础数据：员工基本信息，传感器基本信息，预应力数据等。由于这些数据都是严格的结构化数据，所以我们使用传统的关系型数据库就可以实现所有的数据服务。

3.3 系统安全性保证

鉴于图 3-1 的系统架构，并考虑系统和系统数据的安全性，从组织管理和技术上提出两个方法来保证数据的有效性、安全性。

一是将用户根据权限将用户分为普通用户和管理员。普通用户只具有增（采集预应力数据）和查（查询预应力历史数据）以及其他个人信息修改的权力，以防员工恶意修改、删除数据，造成重大安全事故。管理员具有增删查改所有功能。

二是无论 client 端还是 browser 端，在与服务器通信时，只要涉及账号密码、预应力数据等重要数据就进行加密。

现在网络上用的最多的三个加密算法分别是 MD5，RSA 和 DES^[26]。DES 是一种简单高效的加密算法，它加密速度快但是容易被破解，适合大量数据需要加密的场景。而 RSA 算法是公认的运用最广泛、最有影响力的非对称加密算法^[27]。它算法理论基础是大质数相乘后的因式分解问题。至今尚没有人能破解密钥长度超过 1024 位的 RSA 算法。由于 RSA 算法涉及复杂的大数运算，所以加密解密效率相对较低，适合少量数据加密的场合。在我们这个只需要密码和预应力数据加密的场景十分符合，所以我们选择 RSA 加密方法。加解密的过程（以登陆为例）如图 3-3 所示。

当员工点击登陆以后，client 端或 browser 端先向服务器发送一个请求想服务器获取 RSA 公钥，服务器端随机产生一个 1024bits 的密钥对，私钥仅有服务器端知道，公钥通过 response 返回，client 端或 browser 端将用户密码用返回的公钥进行加密，然后将用户名和已加密的密码再次发送验证登陆，服务器端将密码解密，查询数据库判断是否正确并返回最终结果。

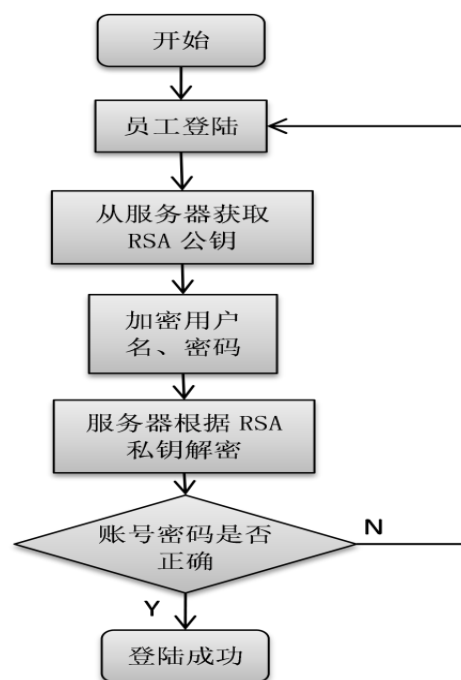


图 3-3 登陆验证流程图

通过抓包软件我们可以验证请求的执行顺序和加密情况。如图 3-4:

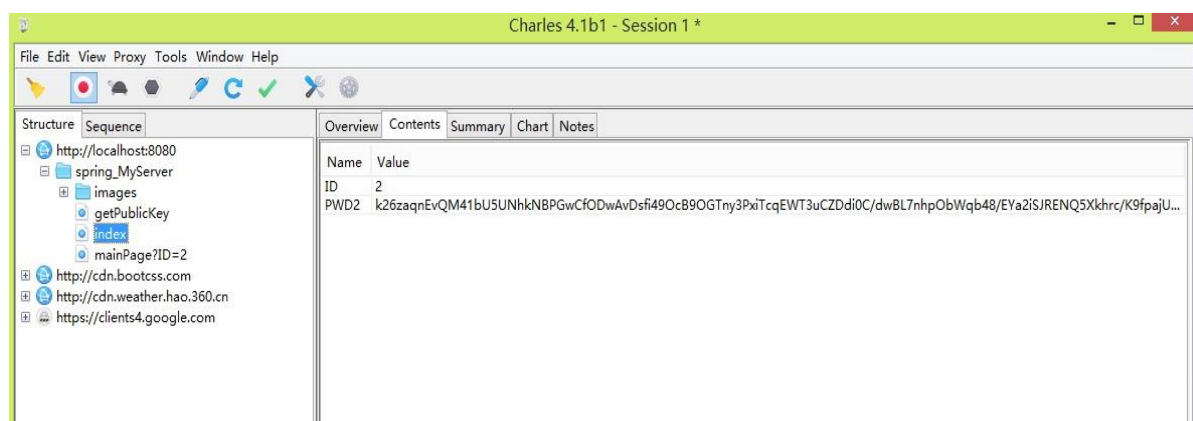


图 3-4 登陆验证流程图

当用户点击“登陆”以后，先发送了一个 `getPublicKey()` 的 GET 请求。获得公钥以后，再发送了一个 POST 请求，在图中右边可以看到，原来 6 位数密码 PWD2 已经变成了一长串字符串，即加密成功。服务器端验证密码正确后，最后跳转到网站主页 `mainPage`。

3.4 本章小结

本章先分析了目前在预应力采集管理上存在的不足，然后根据这些不足完成了系统的需求分析。并在此基础了设计了一个基于 C/S 和 B/S 混合结构的系统，介绍了设计该系统的原因以及每个部分的功能。最后一节介绍了本系统的安全保障：RSA 加密。

第四章 大型预制构件预应力采集管理系统

4.1 App 端

App 端是预应力信息的来源,实现了在线/离线预应力检测、预应力查询、天气查询、定位服务等功能。App 端使用 WPF 框架实现,它是微软新一代的图形桌面 API,具有丰富而强大的 UI 控件,尤其是在 3D 图形和动画开发等方面十分强大。它界面设计与后端逻辑设计分离的规范有利于提高开发的效率。

4.1.1 开发环境

前面已经介绍过 App 端开发语言为 C#, 所以开发环境选择 Visio Studio 2015。作为新版本的微软开发工具, VS2015 支持 .NET Framework 4.5 以上包括 Win10 通用应用在内的所有软件开发。在开发 WPF 时提供大量的 UI 控件用于开发,并且可以设置不同环境进行调试,十分强大便捷。而且 VS2015 对 VS 之前的版本也具有很好的兼容性。

4.1.2 工作流程

App 端工作流程如图 4-1 所示。首先需要用户账号密码登录。登录后可以选择“预应力检测”或“预应力查询”界面,分别实现预应力检测数据或者预应力查询的功能。

如果选择预应力检测功能,则需要先选择串口号、波特率等参数然后打开串口实现串口通信, App 实现了串口自动识别功能,可通过“自动检测”功能试下串口的自动选择。串口选择成功并打开以后就处在工作状态,只要 RFID 读写器读到数据 App 端就能实时显示。读到数据后可以通过“分析数据”功能从位数、字符、ID 值等方面去分析读到的数据是否正确,如果数据正确将从数据库读取预应力转换系数等参数计算该钢绞线的预应力并显示数据值、数据状态、以及传感器的其他基本信息。并由员工选择保存还是丢弃数据。

如果选择预应力查询功能,可以输入要查询到传感器的 ID 并选择查询的起止时间,也可以输入要查询的线路名并输入起止的传感器编号,分别从时间维度微观展示预应力随时间变化情况或者从空间维度宏观展示某条线路不同传感器的预应力变化情况。并且还可以选择折线图、柱状图或者饼状图等不同的图形显示不同的统计信息。

除了如图 4-1 所示的功能外, App 端由于很有可能在网络很差的环境下工作,所以需要离线功能,前一章分析过这是我们选择 C/S 结构的重要原因。它的工作流程很简单:开始—离线登录—串口选择—检测数据—保存至本地—结束。离线功能可以看作基本功能“阉割”版,由于没有网络所以无法实现数据校验、入库以及查询等功能。

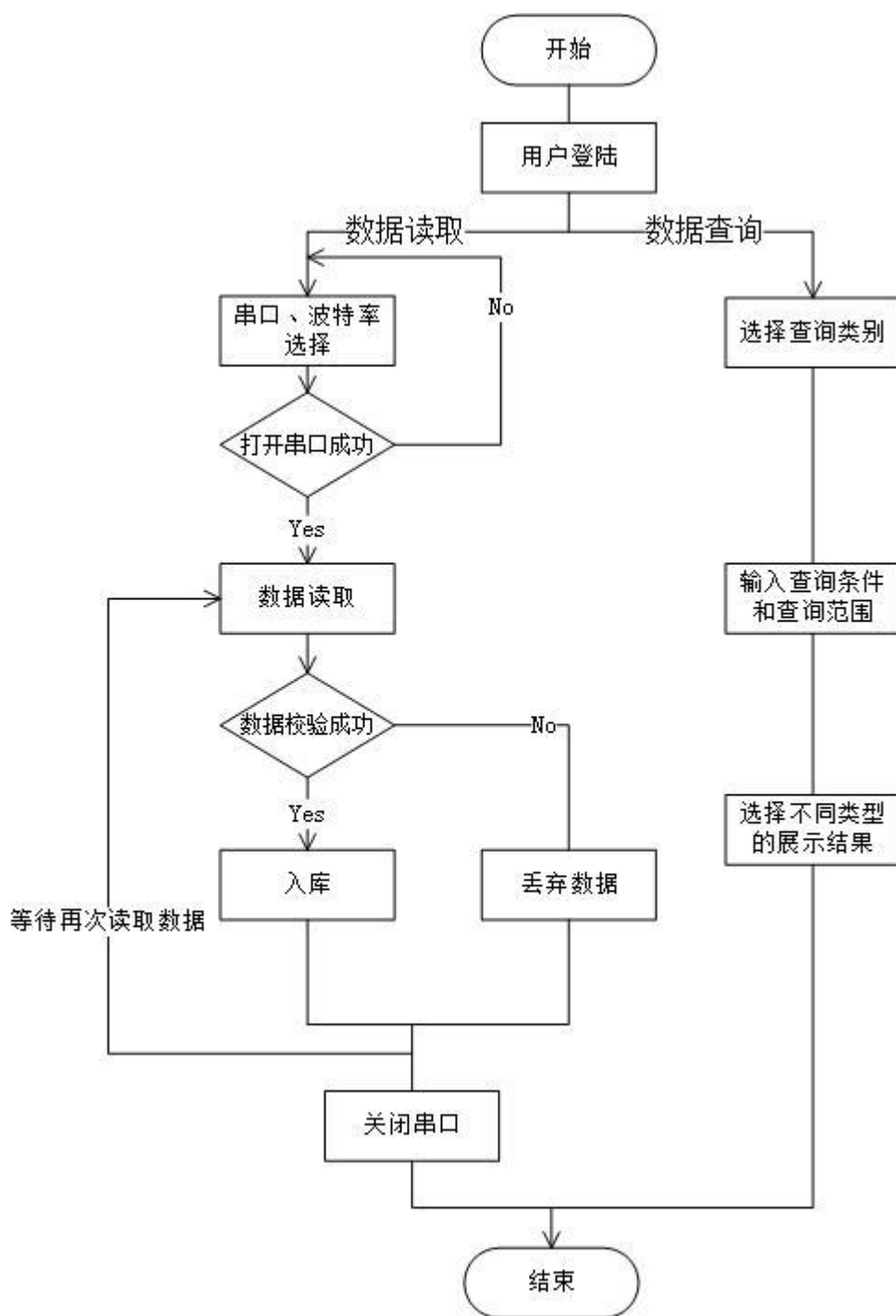


图 4-1 App 端工作流程图

4.1.3 界面、接口与插件

I. 登陆页

图 4-2 为大型预制构件采集管理系统 App 端的登陆界面。在登陆界面输入用户的工号和密码就可以实现登陆。除了正常的登陆功能以外，还提供了“离线登陆”的功能以便在室外没有网络的时候依然可以采集数据并离线储存。界面提供的“记住密码”功能，类似浏览器的 cookie，以加密的方式储存在本地，所以用户不用每次都输入用

用户名密码。



图 4-2 App 端登陆界面

II. 首页



图 4-2 App 首页

首页如图 4-2 所示。App 端所有的界面都分为类似三个模块，左边菜单栏，右边为主界面，最上面还有滚动的通知公告栏。

通知公告栏的“预应力检测”、“预应力数据查询”、“离线数据处理”分别对应 App 端的检测查询以及离线数据的上传或删除。

界面最上方滚动播放当前的时间与公司最近的通知公告。在有网的情况下可以点

击具体的通知跳转到浏览器查看通知详情。

主页上显示了天气与用户所在地。所在地的获取是利用了百度地图开发者工具中开放的 Web 服务 API，其中有一个“普通 IP 定位 API”，只要发送类似“<http://api.map.baidu.com/location/ip?ip=xx.xx.xx.xx&ak=您的AK&coor=bd09ll>”的请求，返回参数就会把你所在的位置（一般为城市级别），返回的位置信息包括经纬度、省等信息。URL 中 ip 即当前上网的 IP，ak 为用户注册后的校验码，coor 为坐标系。

天气的信息获取利用了中华万年历的公共 API。根据上一步获得的城市名，发送“http://wthrcdn.etouch.cn/weather_mini?city=城市名”的 URL 请求。以 json 格式得到本城市一周的天气信息，接口数据如下：

```
{
  "data": {
    "yesterday": {
      "date": "6 日星期二",
      "high": "高温 4℃",
      "fx": "北风",
      "low": "低温 -2℃",
      "fl": "",
      "type": "晴"
    },
    "city": "杨浦区",
    "forecast": [
      {
        "date": "7 日星期三",
        "high": "高温 7℃",
        "fengli": "",
        "low": "低温 -1℃",
        "fengxiang": "南风",
        "type": "晴"
      },
      .....//此处省略一周天气信息
    ],
    "ganmao": "天气寒冷，昼夜温差极大且风力较强，易发生感冒，请注意适当增减衣服，加强自我防护避免感冒。",
    "wendu": "4"
```

```
    },  
    "status": 1000,  
    "desc": "OK"  
}
```

然后对应时间、温度、风向等属性，需建一个 WeatherInfo 类用来表示天气信息。

```
public class WeatherInfo  
{  
    public string week { get; set; }  
    public string date { get; set; }  
    public string high { get; set; }  
    public string low { get; set; }  
    public string direction { get; set; }  
    public string type { get; set; }  
    public string lever { get; set; }  
    public string icoPath { get; set; }  
}
```

WPF 框架采取的是 MVVM 的设计模式，也就是数据驱动 UI 的一种方式。UI 控件在表示层通过数据绑定 (Data Binding) 展示相对应的数据。我们将后台的 WeatherInfo 类与前端的样式模板绑定。当我们解析完 json 数据生成 WeatherInfo 以后，前端就会显示相对应的天气信息了。后台刷新数据后前端也会有相应的变化。

III. 预应力检测



图 4-3 预应力检测界面

预应力检测界面如图 4-3 所示。左边是串口通信模块，可以通过手动或者自动检测的方式选择串口号，并且在选择正确的波特率、数据位等条件后，就可以正确解码读写器发送过来的数据。右边为具体数据分析界面，点击“分析数据”按钮就可以校验数据长度、范围、ID 是否存在，如果正确则会自动向 Web 服务器发送请求获取转换系数并计算最新预应力值。然后根据预应力值范围、传感器信号强度等给传感器或预应力打上“正常”、“偏大”、“偏小”、“失效”等标签。点击“保存数据”按钮就可以将最新的预应力数据保存到数据库服务器。

IV 预应力数据查询



图 4-4 预应力数据查询图 a



图 4-5 预应力数据查询图 b

图 4-4 和 4-5 所示为预应力数据查询界面。查询条件可以选“传感器 ID”或者“传感器线路”，分别从时间上或者空间上展示预应力的情况。

visifire 是 Visifire 公司开发的一套可视化数据软件。它支持快速创建 WPF 和 Silverlight 图标。API 单一简单，但是图形效果炫酷丰富，图表类型包括 2D/3D 类型的柱状图、饼形图、雷达图、漏斗图、泡沫图、K 线图 etc，并支持图形的实时更新。Visifire 的使用也很简单，只需要在工程中下载导入 WPFVisifire.dll 即可。

通过查看预应力数据随时间的变化折线图，就可以知道预应力的趋向走势，传感器的总体部分情况等。

V 离线数据处理



图 4-6 离线数据处理

若 App 是工作在离线模式，就无法完成图 4-1 中的校验和上传等功能。所以离线状态采集页面采集到数据后会不加校验的全部保存在本地数据库。当第二次打开 App 并且是工作在线模式时，通过点击“离线数据处理”，可以加载展示尚未上传的数据。然后点击“上传”按钮，就可以继续完成接下来的校验、上传等流程。校验成功有效，就将数据入库，负责提醒保存失败，并删除相应数据。

4.2 Web 端

Web 端是整个系统的信息管理、展示端，也是最主要的人机交互渠道。Web 前端的搭建我们使用了 Bootstrap 框架。Bootstrap 框架响应式网页的设计理念，在移动互联网技术飞速发展的今天可以大展身手^[28]。Bootstrap 框架最核心的三个技术分别是：弹性化，媒体查询，流式布局^[29]。弹性化指的是图片大小和页面内容会根据不同的设备

分辨率自动调整布局。媒体查询是实现响应式开发的关键。先定义不同的 CSS 样式表，然后通过媒体查询在不同的设备使用不同的页面效果。流式布局是指外层容器会根据显示尺寸伸缩而伸缩以适应不同的显示设备。利用 Bootstrap 进行前端页面开发，其完善的文档和手册，丰富的 UI 和插件，可以大大加快页面的开发速率。

4.2.1 开发环境

考虑到服务器端用 Java 语言开发使用的 IDE 是 Eclipse，所以前端也是用 Eclipse，它是一个十分强大的开发工具，通过不同的插件可以实现 Java、Python、Android 的开发。我们用最新的 Eclipse-Jee-Oxygen 作为前端以及服务器的开发工具，使用 chrome 作为调试浏览器。

4.2.2 功能需求

第三章中对 Web 端总体的功能需求已做出分析：Web 端需要实现除预应力信息采集以外的所有查询、管理功能。在这细化明确 Web 端需要实现的具体功能：

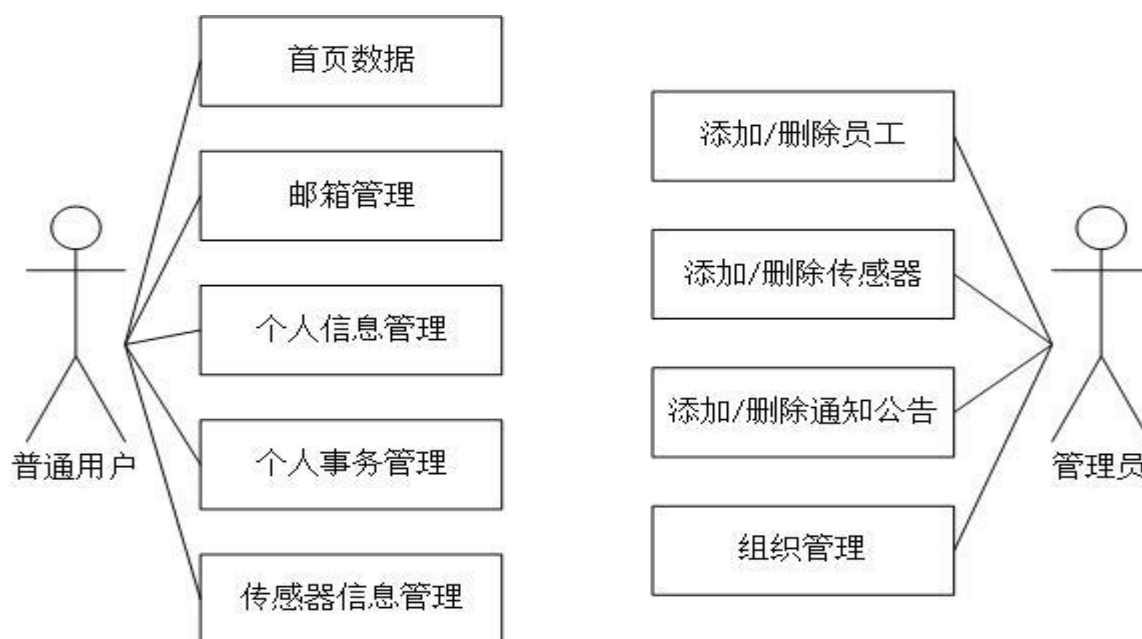


图 4-7 网站用例图

如图 4-7 所示，我们用网站用例图来展示前端需要实现的功能。为了保证系统的数据安全性，防止任意的删除与篡改，根据权限和身份的不同，参与者分为普通用户与管理员。首页展示的数据有通知公告栏、天气、日历各类各类统计数据。邮箱管理用于实现企业内部邮件的发送接收等操作。个人信息管理用于个人信息与登陆密码等。个人事务实现了进度表日程表等功能。传感器信息管理为传感器和预应力的信息的查新。而管理员则负责传感器、员工、通知公告的添加与删除，以及公司、网站的组织管理。

4.2.3 Bootstrap 的引入与应用

Bootstrap 的搭建十分简单，它是一个开源的框架，只需要去官方网站 <https://getbootstrap.com/docs/4.0/getting-started/download/> 下载最新版 Bootstrap，解压，然后将相应 HTML、CSS、JS 文件、图片等加入工程。

引入 Bootstrap 需在相应文件下添加如下代码：

```
<link rel="stylesheet" href="css/style.css"/>
<link rel="stylesheet" href="js/bootstrap/dist/css/bootstrap.css" >
<script src="js/bootstrap/dist/js/bootstrap.min.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

其中前三行代码的作用是引入 Bootstrap 的 CSS 样式表和 JS 文件。

最后一行代码是实现响应式布局的关键代码，它设定了页面的宽度是本设备的宽度，而不是某个固定值。通过媒体查询可以从 style.css 文件中选择最合适的层叠样式来匹配当前分辨率的显示器。

在 CSS 文件中，根据市场上比较常见的一些设备的屏幕大小，我们定义了四种显示样式。四种屏幕分别是大屏幕（笔记本和普通显示器，屏幕像素大于等于 1200 像素），中等屏幕（笔记本和普通显示器，屏幕像素介于 980 像素和 1199 像素），小屏幕（平板电脑，屏幕像素介于 768 像素和 979 像素），超小屏幕（手机，屏幕像素小于 768）。媒体查询实现部分代码如下：

```
@media (min-width: 1200px){
    .cl-sidebar .cl-vnavigation{
        display: block !important;
    }
}
@media (min-width: 980px) and (max-width: 1199){
    .cl-sidebar .cl-vnavigation{
        display: block !important;
    }
}
@media (min-width: 768px) and (max-width: 979px) {
    .butstyle {
        width: 30.9%;
    }
    .butstyle:first-child {
        margin-left: 0;
    }
}
```

```
    }  
    .navbar-default .navbar-brand {  
        background-position: 0 11px;  
        height: 50px;  
        width: 55px;  
        padding-left: 26px;  
    }  
}  
@media (max-width: 767px) {  
    #cl-wrapper {  
        padding-top: 0;  
        display: block;  
    }  
    .cl-mcont {  
        padding: 15px;  
    }  
    /*nav bar*/  
    .navbar {  
        padding-right: 10px;  
    }  
}
```

由代码可知，对于不同的分辨率，设置了不同的长宽比、布局等，使得页面在不同设备都能以最佳的方式呈现。

4.2.4 页面、接口与插件

Bootstrap 的页面布局是基于流式布局实现的，而流式布局实现的关键是栅格系统。栅格系统通过一系列规定的行和列来实现页面元素与容器的布局^[30]。所以整个网页的风格为扁平化、模块化。

页面总体设计采用左边导航栏右边主界面的经典界面。左边的导航栏展示了六个基本功能：首页，邮箱，个人管理，组织管理，传感器管理、预应力管理。

I. 首页

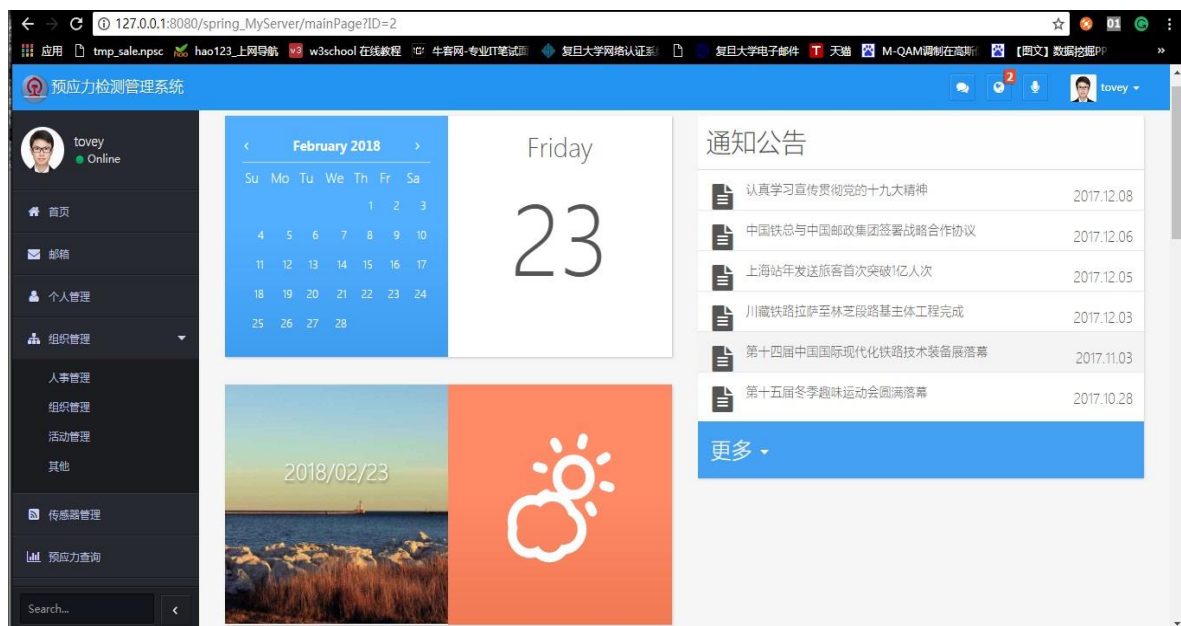


图 4-8 Web 首页 1



图 4-9 Web 首页 2

Web 登陆界面与 App 端类似，此处不再介绍。

首页如图 4-8 所示，左边是导航栏，右边是主界面。根据媒体查询前端自动选择了一种最好的呈现方式。当我们调整屏幕的宽度如图 4-9，网页会自动将左边的导航栏自动缩小为只显示小图标，网页依然能以最佳的视觉呈现给用户。

首页是整个网站的入口，有各子项目的快照、总体数据展示以及通知公告等模块。设置栅格系统将主界面分为两列多行，每一块位置放置不同的功能，具体有日历模块、

通知公告栏、天气模块、预应力变化折线图、线路传感器总数统计图、传感器状态饼状图等各功能和数据的入口。其中天气模块信息来源采用与 App 端相同，都是通过中华万年历的天气开放接口“http://wthrcdn.etouch.cn/weather_mini?city=城市名”获取实时天气信息，再通过前端渲染展示。

II. 传感器管理页面

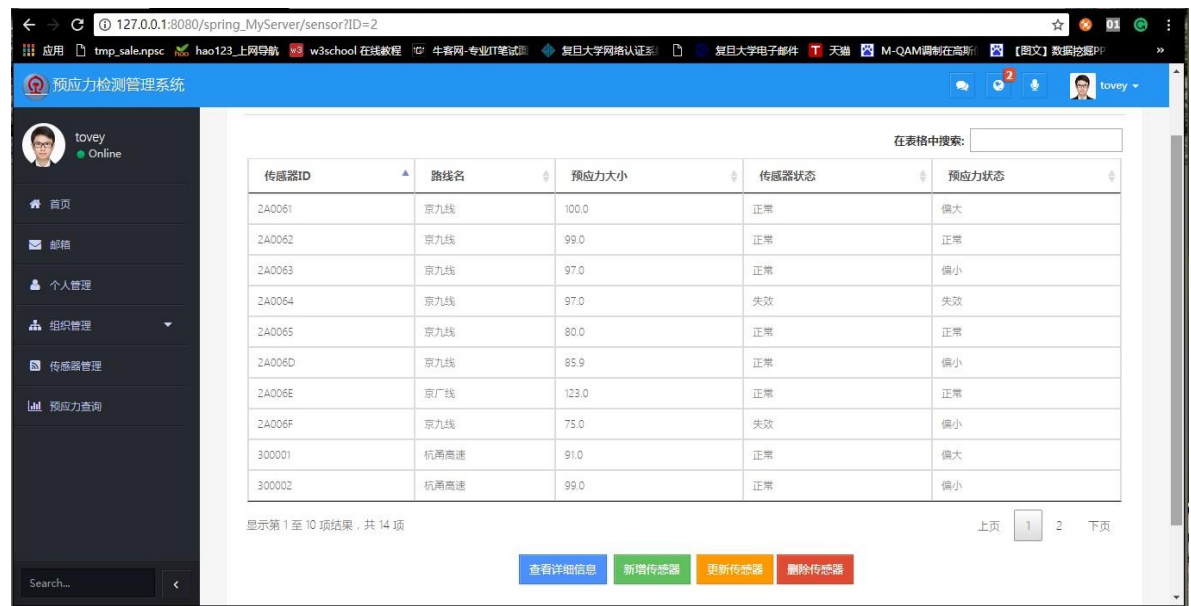


图 4-10 Datatables 表格展示图

包括如图 4-10 所示的传感器管理、组织管理中的人事管理在内的所有表格展示，我们使用 DataTables 插件。DataTables 插件是一个基于 jQuery 的开源表格插件。它功能强大，具有自动分页、即使搜索和排序、支持 jQuery UI、Bootstrap 等不同主题、动态创建表格、具有丰富的 API 等众多优点。

DataTables 的使用非常简单，只需引入相应的 CSS 和 JS 文件即可，代码如下：

```
<link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css"/>
<script type="text/javascript"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
```

若是静态加载表格，只需以下三行代码即可初始化表格插件了：

```
$(document).ready(function(){
    $('#myTable').DataTable();
});
```

但是如果需要动态加载表格，则需要在初始化时与后台数据库交互，指定数据源并渲染，如下：

```
table = $('#datatable').DataTable({
```

```

lengthMenu: [[10, 25, 50, -1], [10, 25, 50, "All"]],
  pageLength: 10, //设置每页显示长度
  data: sensors, //设置数据源
  language: {
    //表格设置，此处省略
  },
  lengthChange: false,
  columns: [
    {data: "sensor_id", defaultContent:""}, //此处设置每列的数据源
    {data: "railway_name", defaultContent:""},
    {data: "stress_recent", defaultContent:""},
    {data: "sensor_state", defaultContent:""},
    {data: "stress_state", defaultContent:""},
  ],
  select: true
});

```

通过 columns 设置每一列的数据源，sensor_id、railway_name 等与传感器 ID、线路名、预应力最新值、传感器状态、预应力状态一一对应。

对于信息的具体增删查改，我们使用 Bootstrap 的弹出框插件 Popover，使用弹出框嵌套 datatable 的模式，利用 Ajax 实现异步刷新，既突出了重点又可以不用跳转界面。如图 4-11 所示。

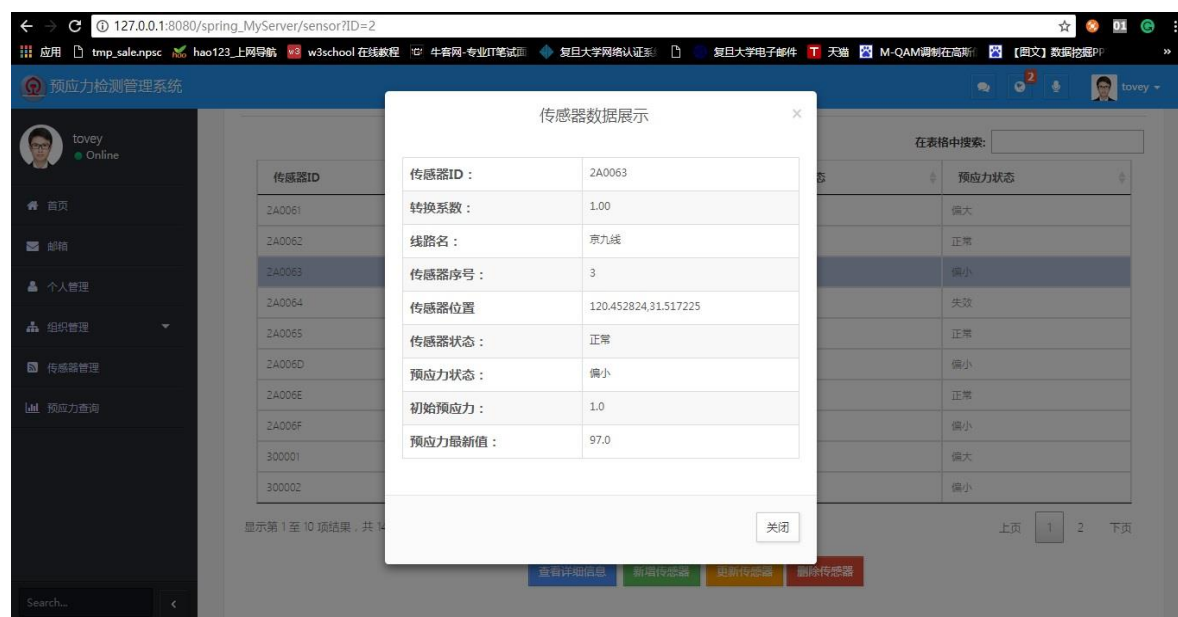


图 4-11 具体数据展示图

III. 预应力查询页面

预应力查询页面涉及到数据可视化。Flot 是用 Javascript 实现的 jQuery 绘图插件。由于是纯 JavaScript 代码实现，所以可以在 IE6+/Firefox2+/Safari3+/Opera9.5+/Chrome 等主流浏览器上运行。使用 Flot 插件只需在 HTML 中加入以下代码：

```
<script type="text/javascript" src="js/jquery.flot/jquery.flot.js"></script>
<script type="text/javascript" src="js/jquery.flot/jquery.flot.pie.js"></script>
```

就可以支持常见的条形图、折线图、饼状图的绘制。



图 4-12 预应力页面查询图

如图 4-12，我们可以通过输入筛选条件，从时间（按时间查询）和空间（按传感器序号查询）上来查看预应力的变化走势，从而分析预应力松弛与时间和空间分布的关系。将鼠标放在相应的点上，会自动显示具体的预应力数据。此外还有传感器状态扇形图、预应力状态扇形图、铁路流量柱状图等不同形状的可视化图表。

IV. 其他插件

为了实现更友好的交互界面，UI 呈现更加和谐一致。在网页中也使用了大量 jQuery 的其他插件。如首页的天气，我们使用了 jQuery 的 jQclouds 插件，它使得我们开发动态的天气动画非常简单；日历使用了 jQuery 的 DateTimePicker 插件；JS 原生的 alert() 插件会使周末变暗并且与网页总体风格不一，我们使用简洁、小巧又会自动消失的 gritter 插件。此处不再一一展示相应的页面。

4.3 服务器端

4.3.1 开发环境

服务器开发语言 Java。可以在 Oracle 官网下载最新版的 JDK8 安装 Java 语言。Sun 公司的 Eclipse Enterprise Edition 是目前最常用的 Java 服务器端开发工具。所以我们选择 Eclipse Jee Mars 作为开发环境。

4.3.2 工作流程

SpringMVC 是一个基于 Java 语言、按照 MVC 架构设计的轻量级 Web 应用框架^[31], 它是 Spring 七大模块之一, 具有 Spring 所有的优点。SpringMVC 主要有五大组件, 分别是前端控制器(DispatcherServlet), 映射处理器(HandlerMapping), 控制器(controller), 视图解析器(ViewResolver)和视图(View)^[32]。

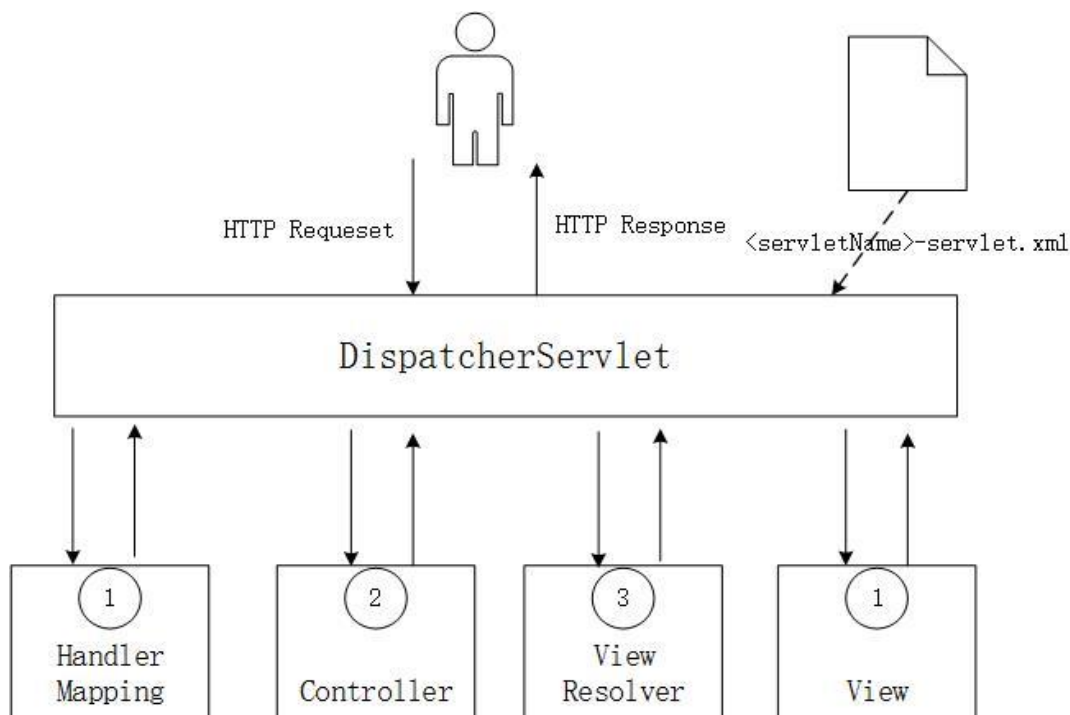


图 4-13 SpringMVC 工作流程图

当用户（浏览器）发送一个 HTTP 请求，Web 服务器接受请求并判断请求的 URL 是否和 XML 文件中规定的需要拦截的 URL 匹配。如果不匹配，由默认 Servlet 处理；如果匹配，Web 容器将请求转发给 DispatcherServlet。然后流程如 4-13 所示：

- (1) DispatcherServlet 收到请求后转发给 HandlerMapping, HandlerMapping 根据配置信息以及 URL 告诉 DispatcherServlet 应该调用哪个 Controller。
- (2) DispatcherServlet 将请求发送给相应的 Controller。Controller 判断是 GET 或 POST 请求调用相应的方法并设置模型数据,并返回视图名称。

- (3) ViewResolver 接受视图名称,并根据配置解析找到真正的视图文件(JSP 或 HTML 文件等),并传递模型数据。
- (4) View 根据模型数据渲染最终要呈现的视图,返回给 DispatcherServlet, DispatcherServlet 在返回给用户(浏览器),用户得到最后的结果。

4.3.3 服务器配置

服务器要完成 4.3.2 节所示的一整套工作流程,有两个重要的配置文件必不可少: Web.xml 和<servletName>-servlet.xml。其中第一个配置文件是针对整个工程的初始化配置:如欢迎页面, servlet 的映射及命名, filter 等。第二个文件是针对 DispatcherServlet 的配置文件,如 ViewResolver 的映射规则,资源和驱动的配置等都在这里完成。只有在 Web.xml 中正确配置 DispatcherServlet,才能让整个程序正确跑起来,核心代码如下:

```
<servlet>
    <servlet-name>dispatcherServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/config/applicationContext.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>dispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
</servlet>
```

该代码引入了 DispatcherServlet,并用 /WEB-INF/config/ 文件夹下的 applicationContext.xml 完成初始化。applicationContext.xml 即上文中的 <servletName>-servlet.xml, <servletName>-servlet.xml 为不进行配置时默认的名字。<load-on-startup>1</load-on-startup>表明启动优先级最高,容器启动时马上初始化。最后的<url-pattern>/</url-pattern>表明拦截一切 URL。

applicationContext.xml 配置信息部分如下:

```
<!-- 获取配置文件 -->
<bean id="config"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="locations">
```



```

    <list>
      <value>/WEB-INF/config/db-config.properties</value>
    </list>
  </property>
</bean>
<!--连接数据库 -->
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
  <property name="driverClassName">
    <value>${db.dirverClass}</value>
  </property>
  <property name="url">
    <value>${db.url}</value>
  </property>
  <property name="username">
    <value>${db.username}</value>
  </property>
  <property name="password">
    <value>${db.password}</value>
  </property>
</bean>
<!--定义视图 通过 internalResourceView 来表示 使用的是 Servlet/jsp 技术-->
<bean id="viewResolver"
  class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/jsp/" />
  <property name="suffix" value=".jsp" />
</bean>

```

PropertyPlaceholderConfigurer 类的作用是管理属性配置文件，db-config.properties 文件即该类的值，在文件中配置了数据库连接的基本信息。所以\${db.url}，\${db.username} 获得数据库的用户名密码等，实现数据库的自动连接。

<property name="prefix" value="/WEB-INF/jsp/" /> 和 <property name="suffix" value=".jsp" />这两行代码设定了 viewResolver 的 Mapping 规则，当 controller 返回字符串“abc”时，viewResolver 会自动去/WEB-INF/jsp/文件夹下寻找 abc.jsp 文件。

4.3.4 MVC 编程

SpringMVC 是一个 Model-View-Controller 的框架，每一层专心做自己的事，这样

的架构开发起来层次清晰维护方便。

SpringMVC 的模型层 (Model) 由简单 Java 对象 (Plain Ordinary Java Object, POJO) 组成。一般与数据库中的相应表一一对应。

比如本系统中最重要的两个对象为员工和传感器。他们的对象类为:

```
Public class User {  
    private Integer employee_id;  
    private String employee_name;  
    private String employee_pwd;  
    private String gender;  
    private String phone;  
    private String birthday;  
    private String department;  
    private String email;  
    private String address;  
    private String lever;  
    //相应的 getter、setter 方法  
}  
  
Public class Sensor {  
    private String sensor_id;  
    private String conver_radio;  
    private String railway_name;  
    private String sensor_location;  
    private String sensor_state;  
    private String stress_state;  
    private String stress_init;  
    private String stress_recent;  
    private Integer sensor_SN;  
    //相应的 getter、setter 方法  
}
```

每一个私有变量分别对应数据库相应的字段。

视图层 (View) 用于处理数据的显示, 将最后的界面在浏览器呈现出来就是它的工作, 用的最多的就是 JSP 技术。页面的布局遵循最新的 HTML5 的规范。此处不再展示。

控制层 (Controller) 负责接收请求, 并将任务分发给 service 层, 最后负责将结果

封装成 model 返回给视图。

我们在 applicationContext.xml 文件中加入代码：<context:component-scan base-package="pkg_controller"/>，就可以实现自动扫描装载 pkg_controller 包路径下要用到的 bean。

通过以下方式对不同的 URL 调用不同的函数最后返回结果给 ViewResolve。

```
@RequestMapping(value="/getSensor")
Public String yourMethod (Model model,  HttpServletRequest request){
//...
return "sensorList";
}
```

4.4 数据库

数据库的选择是否恰当与系统的运行速率有着密不可分的关系。考虑到本系统为公司内部系统，并发量、数据量不大的特点，我们选择 MySQL 作为数据库。MySQL 是 Oracle 公司旗下的一款轻量级关系型数据库，与其他类型的数据库相比其运算速度较快，并且具有较高的可靠性和安全性^[33]。根据数据库设计的三大基础原则^[34]和本系统的功能需要，设计数据库表格最主要有三张：用户信息表，传感器信息表和预应力记录表。

1. 用户信息表 (user_tb)

| 字段名 | 数据类型 | 长度 | 允许空 | 唯一 | 主键 | 默认值 | 说明 |
|---------------|---------|----|-----|----|----|-----|-------|
| employee_id | int | 5 | × | √ | √ | 自增 | 员工 ID |
| employee_name | varchar | 20 | × | × | × | 无 | 员工姓名 |
| employee_pwd | varchar | 20 | × | × | × | 无 | 员工密码 |
| gender | enum | | √ | × | × | 男/女 | 员工性别 |
| phone | char | 11 | × | √ | × | 无 | 联系电话 |
| birthday | char | 8 | √ | × | × | 无 | 员工生日 |
| department | varchar | 20 | × | × | × | 无 | 所在部门 |
| Email | char | 50 | × | √ | × | 无 | 电子邮箱 |
| address | varchar | 20 | √ | × | × | 无 | 员工住址 |
| lever | int | 2 | × | × | × | 1 | 员工级别 |

用户信息表用于保存员工的个人信息。员工 ID 为自增主键；性别 gender 为枚举类，默认值为男或女；lever 为员工的级别，默认新员工为 1 级。

2. 传感器信息表 (sensor_tb)

| 字段名 | 数据类型 | 长度 | 允许空 | 唯一 | 主键 | 默认值 | 说明 |
|---------------|---------|----|-----|----|----|-----------------|--------|
| sensor_id | char | 6 | × | √ | √ | 无 | 传感器 ID |
| conver_ratio | double | 6 | × | × | × | 无 | 转换系数 |
| railway_name | varchar | 10 | × | × | × | 无 | 所在线路名 |
| sensor_lng | varchar | 10 | × | × | × | 无 | 经度 |
| sensor_lat | varchar | 10 | × | × | × | 无 | 纬度 |
| sensor_state | enum | | √ | × | × | 正常/失效 | 传感器状态 |
| stress_state | enum | | √ | × | × | 正常/偏大/ 偏小/失效 | 预应力的状态 |
| stress_init | double | 6 | × | × | × | 无 | 预应力初始值 |
| stress_recent | double | 6 | √ | × | × | 无 | 预应力最新值 |
| sensor_SN | int | 20 | × | × | × | 无 | 传感器序列号 |

传感器信息表用于存储传感器的基本信息。sensor_id 是指传感器 ID, 该 ID 唯一, 在出厂时已经烧写在 EEPROM 里【刘青】。而 sensor_SN 指的是传感器在某条线路部署时的序号, 从 1 开始递增。传感器状态 (sensor_state) 有两种: 正常和失效。正常是指传感器能正常采集预应力信息并与 RFID 读写器正常通信, 失效指的是传感器由于机械故障无法正常采集或发送预应力信息。预应力状态 (stress_state) 是指该传感器所监控的点的预应力的状态。正常表明预应力在正常范围, 而偏大/偏小指的是预应力与标准范围相比已经略微偏大或偏小, 需要密切关注接下来的变化情况; 失效表示预应力已经严重失常, 需要马上采取应对措施, 防止出现安全事故。

3. 预应力记录表 (prestress_tb)

| 字段名 | 数据类型 | 长度 | 允许空 | 唯一 | 主键 | 默认值 | 说明 |
|------------|--------|----|-----|----|----|-----|--------|
| data_id | int | 64 | × | √ | √ | 自增 | 记录编号 |
| data_value | double | 6 | × | × | × | 无 | 预应力值 |
| sensor_id | char | 6 | × | × | × | 无 | 传感器 ID |
| user_id | int | 4 | × | × | × | 无 | 用户 ID |
| date | date | | × | × | × | 无 | 采集日期 |

预应力记录表用于记录预应力数值, 以及由哪位员工采集, 采集日期等数据。本章表中的 sensor_id 为外键, 对应 sensor_tb 中的 sensor_id, 对应的规则是删除时 “CASCADE”, 更新时 “CASCADE”。本章表中的 user_id 也设置为外键, 对应 user_tb 中的 employee_id, 对应的规则是删除时 “NO ACTION”, 更新时 “CASCADE”。

4.5 本章小结

本章是本文的核心章节，具体介绍了预应力采集管理系统软件各个部分的功能。**App** 端是预应力信息的来源，具体介绍了 **App** 端在线和离线状态不同的工作流程以及 **App** 端所用到的接口和插件。**Web** 端作为主要的人机交互渠道，文中重点介绍了人机交互界面的设计思路风格，**Bootstrap** 其他 **jQuery** 插件的具体应用情况。对于服务器端我们重点介绍了服务器的配置状况，并给出了 **MVC** 各层的核心代码。数据库端给出了数据表的设计思路以及三张核心表的设计图。

第五章 基于改进型 GM (1,1) 模型的预应力预测算法

前几章对预应力的采集和展示做了具体的介绍,本章将进一步对已经采集的预应力数据做分析处理。数据的采集和储存并不是最后目的,只是手段。大数据时代缺的不是数据,激活数据、深入挖掘这些数据的价值才是当务之急^[35]。目前已有很多专家学者为预应力构件或者其它混凝土构件中的预应力的损失模型或预测模型做出了贡献。如在文献^[36]中,吕志涛院士对预应力的长期损失的计算考虑到了构件的收缩徐变、钢筋钢绞线的松弛以及摩擦损失等众多因素,得出了一个长期损失的计算公式,这是工程上经典的计算方式。文献^[37]中,作者也是采用类似的方式,综合考虑混凝土构件、天气、弹性模量等原因得到核电站安全壳的预应力损失公式。这些公式虽然在理论上具有很重要的意义,但是由于在实际应用中,我们不可能将所有的因素都考虑周到,比如在高速公路、预应力桥等应用场景,交通的流量(即对应构件的使用强度)必然是一个影响预应力损耗的重要因素。然而在所有的公式中都没有考虑到这个因素,因为这个因素是比天气更不可预测。

所以要更好的预测预应力的损失,最好的办法是从数据本身出发,去发现预应力损失的规律。虽然不同应用场景、不同环境的预应力预制构件中的预应力变化规律不尽相同,但是对同一个位置传感器采集的预应力数据来说,环境、混凝土构件的物理特性等都可看作固定,故预应力的损失有一定的规律。对于有很多不确定的因素组成的系统,可以用灰度模型(GM (1,1) 模型)理论进行预测分析^[38]。

5.1 GM (1,1) 模型简介

GM (1,1) 模型在工程学、经济学等由众多不确定因素组成的系统中常被用到。它的建模思想和过程是:设原始非负序列为 $X^{(0)} = (x^{(0)}(1), x^{(0)}(2) \dots, x^{(0)}(n))$, 如果满足级比:

$$\lambda(k) = \frac{X^{(0)}(k-1)}{X^{(0)}(k)}, k = 2, 3, \dots, n \quad (5-1)$$

在区间 $(e^{-\frac{2}{n+1}}, e^{\frac{2}{n+1}})$ 内, 就可以使用 GM (1,1) 模型建模。由于序列 $X^{(0)}$ 的任一数据具有随机性, 可以通过将数据累加来降低随机性, 发现数据规律, 即:

$$X^{(1)}(k) = \sum_{i=1}^k X^{(0)}(i), k = 1, 2, \dots, n \quad (5-2)$$

由 $X^{(1)}$ 生成背景序列 $Z^{(1)} = (z^{(0)}(2), z^{(0)}(3) \dots, z^{(0)}(n))$, 其中

$$z^{(1)}(k) = \alpha x^{(1)}(k) + (1 - \alpha)x^{(1)}(k-1) \quad (5-3)$$

一般 α 取 0.5。

在文献^[39]中, 给出了某处预应力混凝土桥梁的预应力数据的监测值, 如下表所示:

| 节点名称 | 不同年份的节点预应力值/MPa | | | | |
|-------|-----------------|------|------|------|------|
| | 2013 | 2014 | 2015 | 2016 | 2017 |
| Node1 | 8.09 | 8.07 | 8.04 | 8.03 | 8.01 |
| Node2 | 5.12 | 5.1 | 5.07 | 5.05 | 5.04 |
| Node3 | 5.42 | 5.4 | 5.37 | 5.35 | 5.34 |
| Node4 | 7.94 | 7.92 | 7.89 | 7.88 | 7.87 |

表 5-1 预应力随时间变化图

我们取表中的任意一组数据作为原始数据 $X^{(0)}$ ，并计算背景值序列 $X^{(1)}$ ，用 MATLAB 进行曲线拟合，如图 5-2,5-3 所示：

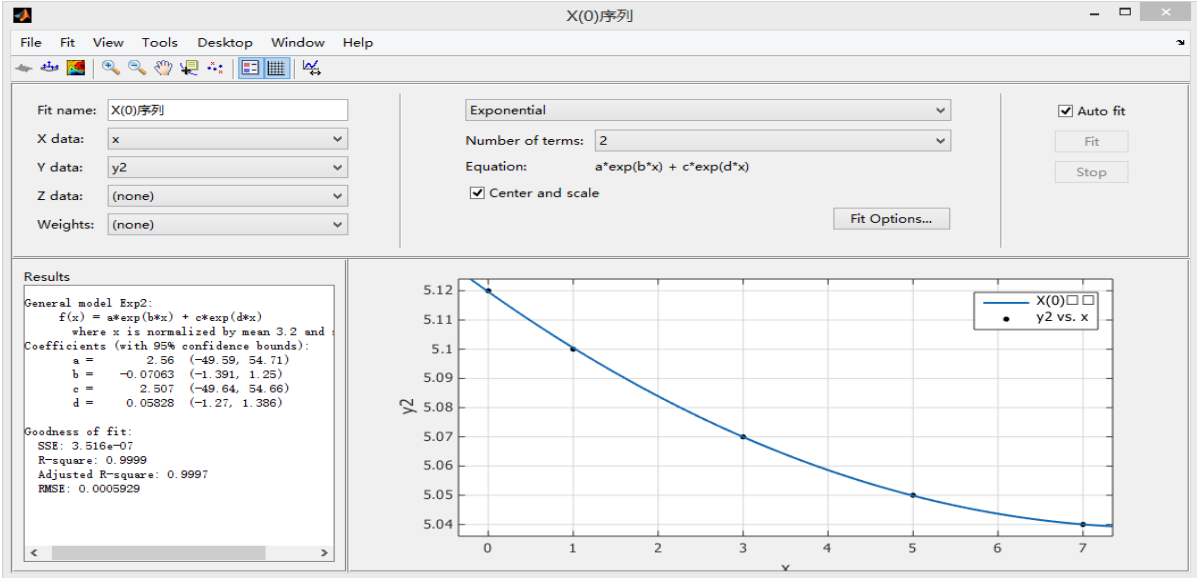


图 5-2 $X^{(0)}$ 序列拟合图

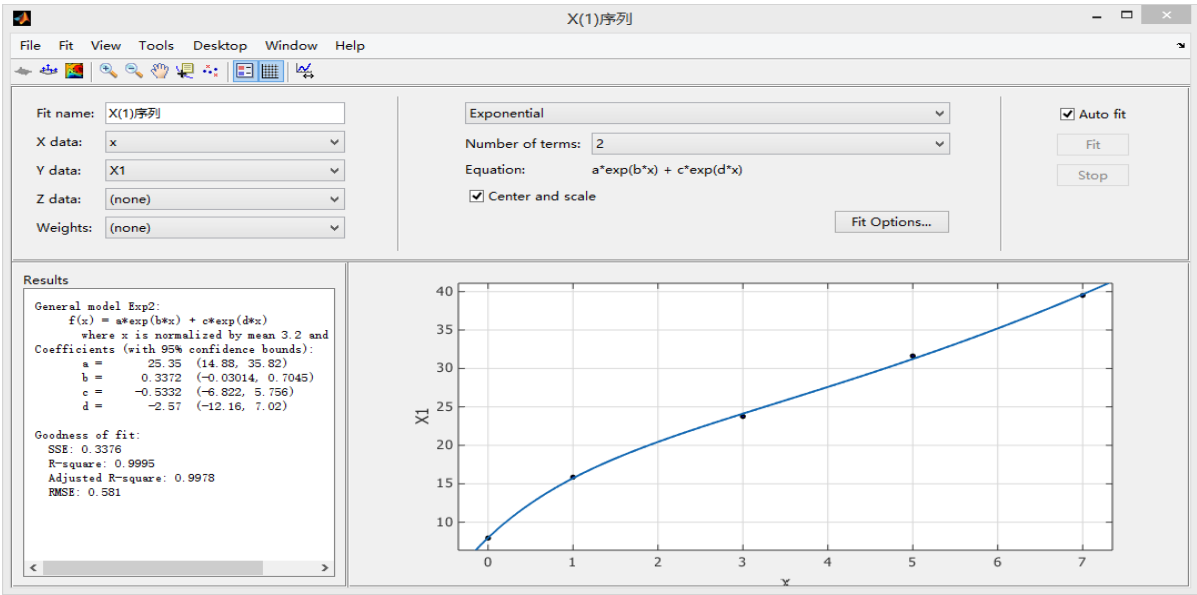


图 5-3 $X^{(1)}$ 序列拟合图

由图可知不论是 $X^{(0)}$ 还是 $X^{(1)}$ 序列规律都是类似指数函数。在其他很多文献资料中预应力也是指数衰减, 故该数据有代表性。接下来我们都用该数据进行验证分析。

由于背景值序列符合指数函数的特点, 故满足白化微分方程:

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \quad (5-4)$$

在离散序列中, 我们可以将微分方程变成差分:

$$x^{(0)}(k) + az^{(1)}(k) = b \quad (5-5)$$

其中参数 a 为发展系数, b 称为灰作用量。故 GM (1,1) 模型重点就是求 a, b 两个

参数的值。令: $c = \begin{bmatrix} a \\ b \end{bmatrix}$, $Y = \begin{bmatrix} x^{(0)}(2) \\ \vdots \\ x^{(0)}(n) \end{bmatrix}$, $B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix}$, 可以用最小二乘法求得 \hat{c} :

$$\hat{c} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = (B^T B)^{-1} B^T Y \quad (5-6)$$

而 $X^{(0)}$ 和 $X^{(1)}$ 的预测公式分别如下:

$$\hat{x}^{(1)}(k+1) = (x^{(1)}(1) - \frac{b}{a})e^{-ak} + \frac{b}{a} \quad (5-7)$$

$$\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k) = (1 - e^a)(x^{(0)}(1) - \frac{b}{a})e^{-ak} \quad (5-8)$$

将公式 (5-6) 中得到的 a, b 的预测值代入就可以得到 $X^{(0)}$ 和 $X^{(1)}$ 的具体值。

5.2 GM (1,1) 模型的优化与改进

灰度模型的精度和准确度与背景序列 $Z^{(1)}$ 以及初始值的选择有重要关系。因为背景序列 $Z^{(1)}(k+1)$ 表示的是在 $[k, k+1]$ 这段时间内 $dx^{(1)}/dt$ 的背景值, 若在 dt 时间内序列变化缓慢, 背景值这么计算是合理的, 但是如果背景值变化剧烈, 这么计算误差就很大了; 此外, 初始值 $x^{(1)}(1)$ 的选择也不是最优, 因为它是一个非累计值, 具有随机性^[40], 我们之所以选择 $X^{(1)}$ 而不是 $X^{(0)}$ 就是希望从累加值发现变化规律, 降低单个数据随机性造成的影响。所以我们要从改变这两个值入手去优化传统的 GM (1,1) 模型。

在文献^[41]中已经提出了新的预测公式:

$$\hat{x}^{(1)}(k+1) = (x^{(1)}(m) - \frac{b}{a})e^{-a(k-m+1)} + \frac{b}{a} \quad (5-9)$$

这里的 m 取值可以根据实际情况选择 1 至 n 中的任意一个值, 当选择 $m>1$ 时就可以抛弃具有随机性的 $x^{(1)}(1)$ 作为初始数据, 我们在这里借用此公式。此外传统的公式: $z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1)$, 之前也分析了, 在相邻数据剧烈变化时会有较大的误

差,所以在这里我们采用加权重的办法,使得背景值的选择更加合理。通过加权的方式就可以考虑到不同序列值变化情况,让背景值的选择更加贴近真实值。在此处提出了一种新的背景值就算公式:

$$z^{(1)}(k) = \frac{x^{(1)}(k)}{x^{(1)}(k) + x^{(1)}(k-1)} x^{(1)}(k) + \frac{x^{(1)}(k-1)}{x^{(1)}(k) + x^{(1)}(k-1)} x^{(1)}(k-1) \quad (5-10)$$

通过改变 $x^{(1)}(k)$ 和 $x^{(1)}(k-1)$ 前的系数,让背景值更加贴近真实的变化情况。

经计算检验,表 5-1 中的数据满足 $(e^{-\frac{2}{n+1}}, e^{-\frac{2}{n+1}})$ 的级比要求,并且现实模型存在众多不确定的参数,故可以使用灰度模型(GM(1,1))算法来进行预测。取表 5-1 中第二条数据对传统 GM 模型和改进型 GM 模型进行准确度分析。

将节点 2 的数据代入公式(5-6)可得 $a=0.0049, b=5.1362$, 公式(5-7)变为:

$$\hat{x}^{(1)}(k+1) = (x^{(1)}(1) - 1048.204)e^{-0.0049k} + 1048.204 \quad (5-11)$$

另 $k=4$, 即可算出 $\hat{x}^{(1)}(5)$ 为 25.3654, 故 $\hat{x}^{(0)}(5)=5.0254$ 。

若选择改进型 GM(1,1) 算法, 则 $a=0.0052, b=5.1422$, 令 $m=2$, 代入公式(5-9)得:

$$\hat{x}^{(1)}(k+1) = (x^{(1)}(2) - 988.88)e^{-a(k-1)} + 988.88 \quad (5-12)$$

另 $k=4$, 即可算出 $\hat{x}^{(1)}(5)$ 为 25.3687, 故 $\hat{x}^{(0)}(5)=5.0287$ 。

对比原值 5.04, 可知改进型 GM(1,1) 算法预测更准确。

用表 5-1 中的前 4 所有数据作为训练拟合数据, 最后一年的数据作为测试数据检验两种算法模型的准确度。结果如以下两表:

| 节点 | 实际值 | 预测值 | 残差 | 相对误差 |
|------|------|--------|--------|--------|
| 节点 1 | 8.01 | 8.0053 | 0.0047 | 0.058% |
| 节点 2 | 5.04 | 5.0254 | 0.0146 | 0.290% |
| 节点 3 | 5.34 | 5.3203 | 0.0197 | 0.369% |
| 节点 4 | 7.87 | 7.8598 | 0.0102 | 0.130% |

表 5-4 GM(1,1) 模型算法预测准确度数据表

| 节点 | 实际值 | 预测值 | 残差 | 相对误差 |
|------|------|--------|---------|--------|
| 节点 1 | 8.01 | 8.0144 | -0.0044 | 0.055% |
| 节点 2 | 5.04 | 5.0287 | 0.0113 | 0.224% |
| 节点 3 | 5.34 | 5.3293 | 0.0107 | 0.200% |
| 节点 4 | 7.87 | 7.8602 | 0.0098 | 0.124% |

表 5-5 改进型 GM(1,1) 模型算法预测准确度数据表

对比以上两表可知, 改进型 GM(1,1) 模型算法来预测预应力值, 比传统 GM(1,1) 模型算法有更高的精度和准确度。在预应力采集管理系统中采用改进型 GM(1,1) 来

预测预应力值将会有更精确的结果。

第六章 总结与展望

6.1 工作总结

随着工程建筑学的发展，现在高铁、高架已经桥梁等众多大型预制构件都选择使用预应力构件，它可以很好的防止预应力构件的提前开裂。但是目前为止对已经投入使用的大型预制构件的预应力变化情况的监控，依然是薄弱的环节。在物联网技术发展成熟的今天，传统行业之前很多无法解决的问题，随着网络、智能传感器等技术的出现都已迎刃而解。本文创新地提出使用物联网的技术，实现对传统用建筑行业的生产和管理实现长期有效监督。

主要完成的工作有：

(1)客观分析了现有的预应力检测存在缺陷和不足。认真对比了物理法、油压法、频率法等传统预紧力检测在精度、长度检测等方面的不足，然后分析了运用无线无源传感器和 RFID 技术进行预应力测量的优势。

(2)对已有的传感器、应变片已经预应力钢绞线进行参数分析，指出不能直接用应变片进行测量；然后对实验室师兄设想的预应力转变装置，进行了严格的数学推导，证明该转变装置可以的正确性，可以将预应力数据转换为电信号。

(3)根据现状和不足，对大型预制构件预应力系统的采集管理系统的需求进行了分析。阐明了本系统采用 C/S 和 B/S 混合模型的原因指出本系统最重要的几个需求是：智能采集、智能展示与分析、提前预警。

(4)仔细分析介绍了预应力采集管理系统的 App 端、Web 端以及服务器的开发环境、工作流程、插件核心技术等方面。App 端采用 C# 的 WPF 框架完成了串口通信助手、天气动画展示、通知公告滚动以及预应力信息展示等功能，并且区分在线和离线模式。Web 端是 Bootstrap 框架完成开发，并且对于我们平时常见的电脑、pad、手机等不同的分辨率定制了不同的显示方式和风格。服务器端分析了 SpringMVC 工作流程并给出了不同层工作的核心配置和代码。

(5)提出了基于加权思想优化背景值序列的改进型 GM (1,1) 算法模型，并用数据检验了该模型的预测准确性，该模型将被用于预应力预测预警。

6.2 研究展望

系统总体上已经满足了预应力信息采集、管理、查询、预测预警等功能，但是由于个人时间和精力有限等原因。整个系统也存在着不足。最主要有两个方面：

(1)在移动互联网时代，随着手机性能的提升，用 Pad 作为移动终端的方式终将被淘汰。并且目前 RFID 读写器与 pad 连接是通过 RS-232 转 USB 连接线连接，所以整个采集系统（Pad-转接线-RFID 读写器）略显笨重。如果我们采用手机作为移动终端，

然后利用蓝牙或者 WIFI 模块让读写器和手机通信，整个系统就会更加轻便和携带。重新改变 RFID 读写器通信模块并设计 Android 端的 App 应用将会是下一步研究方法和趋势。

（2）由于时间有限，整个后台系统依然有很多不足之处。比如一个完整、安全、可拓展的系统，AOP 编程是不可缺少的，主要包括：日志统计、性能统计、异常处理、安全控制等。但是本系统中这些功能并没有加，在后期随着系统服务功能的增加，为了保持系统的稳定，这些功能后期必须一一完善。

参考文献

- [1]. 李建新. 浅谈先张法预应力技术在高速公路桥梁预制构件中的应用[J]. 辽宁交通科技, 2000, 23(6):31-32.
- [2]. 彭仙淼. 高速公路桥梁预应力检测及控制技术研究[J]. 广东交通职业技术学报, 2017, 16(1):35-38.
- [3]. 刘大伟. 预应力锚固系统的检测与监控[J]. 筑路机械与施工机械化, 2002, 19(99):25-26.
- [4]. 丁红岩, 赵思思, 刘鹏, 郭耀华. 简支钢梁自振频率对预应力敏感性试验研究[J]. 石家庄铁道大学学报, 2017, 3(2):1-5.
- [5]. 梁波. 公路桥梁施工中预应力的应用[J]. 路桥管理, 2017,(23):325.
- [6]. R.F.Rinesch. New Ways of Designing Large Refining Vessels And Steelmaking Plants For The LD Process[J]. Journal OF METALS, 1996,18(7):876-880.
- [7]. Feng Lei, Song Xingzhe. Design of the Instrument Soft Panel using Windows Presentation Foundation[J]. 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, Sept. 2015, pp.927-929.
- [8]. Erik Sorensen. Model-View-ViewModel(MVVM) design pattern using windows presentation foundation(WPF) Technology[J/OL]. MegaByte Journal, 2010:2-4.
- [9]. 李夏君. 使用 Bootstrap 实现响应式布局[J]. 信息与电脑, 2017(13):24-25.
- [10]. 韦建波, 韦龙勇. 浅议 bootstrap 框架优缺点[J]. 科技视界, 2017(12):222.
- [11]. 王佳康, 李明, 买尔夏提·尔肯. 基于 SpringMVC 架构的旅游攻略网站前后台的设计与实现[J]. 电脑知识与技术, 2015, 11(8):268-269.
- [12]. 郑洁, 钱育蓉, 范迎迎. 基于 SpringMVC 和 Hibernate 的企业人事管理系统[J]. 电脑知识与技术, 2016, 12(2):15-17.
- [13]. 祝志祥, 陈新, 马光, 陈保安, 韩钰, 邓元, 史永明. 无源传感器的研究与应用进展[J]. 智能电网, 2015, 3(3):191-195.
- [14]. Glynne-Jones P, White N M. Self-powered systems: a review of energy sources[J]. Sensor Review, 2001, 21(2):91-98.
- [15]. Bogue R. Energy harvesting and wireless sensors: a review of recent developments[J]. Sensor Review, 2009, 29(3):194-199.
- [16]. Timo S, Jarmo V. Interrogation techniques for sensors utilizing inductively coupled resonance circuits[J]. Procedia Engineering, 2010(5):216-219.

- [17].谢彦召, 郑振兴, 焦杰. 无源电光式传感器及其进展[J]. 传感器技术, 1999, 18(3):5-7.
- [18].Hu Y, Zhang Y, Xu C, et al. Self-powered system with wireless data transmission[J]. Nano Letters, 2011, 11(6):2572-257.
- [19].刘青, 夏永明, 钱松荣. 带负载匹配耦合谐振电路的能量与数据传输[J]. 太赫兹科学与电子信息学报, 2014,12(6):852-857.
- [20].刘巍. 应变式传感器的原理及对应变片性能的测定[J]. 科技经济市场, 2015, 2:102-103.
- [21].Christophe Carral, Nicolas Charvin. An experimental analysis of PEMFC stack assembly using strain gage sensors[J].International Journal of Hydrogen Energy,18 March 2014,Vol.39(9).
- [22].李巧真, 李刚, 韩钦泽. 电阻应变片的实验与应用[J]. 实验室研究与探索, 2011(4):134-137.
- [23].颜全哲. 预应力钢—混凝土组合梁力学性能研究[D]. 长安大学, 2007.
- [24].刘书成, 胡立栓, 刘景云, 亓呈明, 王育平. 基于 C/S 与 B/S 相结合的企业员工考勤系统[J]. 智能处理与应用, 2017, (4):86-90.
- [25].刘珍. Web 服务器故障的分析与处理[J]. 计算机安全与维护, 2107, (21):87.
- [26].Das Saikat, De Saugatag, Kumar Rahul. Implementation and Comparative Analysis of RSA and MD5 Algorithm[J]. International Journal of Information Security and Privacy, July-Sept, 2012, Vol.6(3), p.1(37).
- [27].司应硕, 杨文涛, 张森. 一种基于 MD5 与 RSA 算法的数字签名系统设计与实现[J]. 新乡学院学报, 2011, 28(1):44-46.
- [28].Pijanowski, Keith. Building responsive Web sites with Bootstrap[M]. MSDN Magazine, 2015, Vol.30(6),p40(5).
- [29].刘志东, 陈天伟. 基于 Bootstrap 的响应式网页设计与实现[J]. 电脑知识与技术, 2017, 13(7):85-87.
- [30].孟洁. 基于 Bootstrap 快速实现网页布局案例研究[J]. 电脑知识与技术, 2017, 13(15):158-159.
- [31].舒坦. 浅析基于 SpringMVC 的高校离退休工作管理系统设计[J]. 系统信息工程, 2017, (9):56.
- [32].赵璘, 王红霞. 基于 SpringMVC+JDBCTemplate 的 Web 系统的研究与应用[J]. 软件工程, 2017, 20(1):5-8.
- [33].黄宁. 基于 MySQL 数据库安全研究[J]. 科学中国人, 2017(6):16-17.
- [34].温立辉. 关系数据库设计原理与分析[J]. 无线互联科技, 2018(1):58-59.

- [35].Commvault. 激活企业数据，释放潜藏价值[J]. 软件和集成电路, 2017(12):14-15.
- [36].周燕勤, 吕志涛. 预应力长期损失计算建议[J]. 东南大学学报, 1997(12):76-80.
- [37].廖开星, 李毅, 孔祥龙, 汤志杰, 遑文新. 核电站安全壳混凝土结构长期预应力预测模型初步研究[J]. 混凝土, 2015(5):140-144.
- [38].郑红超, 黄质宏. 基于灰色理论的预应力锚索抗滑桩预应力损失预测分析[J]. 人民珠江, 2017,38(5):67-69.
- [39].贺志勇, 陈智凯, 易功超. 桥梁预应力损失与刚度变化预测方法研究[J]. 中外公路, 2015, 35(3):131-136.
- [40].董奋义, 田军. 背景值和初始化条件同时优化的 GM (1,1) 模型[J]. 系统工程与电子技术, 2007, 29(3):464-466.
- [41].张大海, 江世芳, 史开泉. 灰色预测公式的理论缺陷及改进[J]. 系统工程理论与实践, 2002(8):140—142.

攻读学位期间发表的学术论文

1. 王志文，钱松荣，《大型预制构件的钢绞线预应力检测与计算》，微型电脑应用，已发表
2. 王志文，钱松荣，《大型预制构件预应力采集管理系统》，微型电脑应用，已录用

致 谢

三年时光转瞬即逝，即将告别复旦、告别最后的校园生活，心中十分不舍。复旦这三年的研究生生涯极不平凡，在这里我认识了很多优秀的师兄师姐和同学朋友，他们都是大神、学霸，他们宽广的知识面、醉心学术的态度值得我学习效仿。感谢他们这三年的陪伴和鞭策，让我有动力和目标去做更优秀的自己。

论文至此已要结束，马上就要走向社会，还是要在这一特别感谢一下这些年帮助、陪伴过我的老师和朋友。感谢钱松荣教授这三年在学术的指导和帮助，钱教授一丝不苟的学术精神和孜孜不倦的育人精神，使我在这三年的研究生受益匪浅，他注重动手能力的培养方式和天马行空的思维跳跃，很大程度上影响了我的学习方式，这将会是受益终生的无形财富。

感谢夏勇民老师在生活中对我们实验室全体同学的照顾。这位早已退休的老师却依然以饱满的投入学生工作中。除学术之外我们遇到的所有问题都喜欢找夏老师帮忙，它为我们安心学术提供了一个可靠的保证。

感谢林春兴、郭慢、刘思豪、高昊、徐晨畅、张小艳等实验室的小伙伴们这三年的陪伴和帮助。尤其是感谢张聪和白东学长，感谢他们在我项目、学习遇到困难时的帮助和指导，和他们一起探讨学术探讨编程是我实验室最开心的时光。感谢我的室友袁辽、张荣升、张成，我们寝室同学关系特别好。一起生活、一起旅游、一起吃火锅的日子是我们一起度过的最美好的时光。

还要感谢我的好朋友袁腾，这么多年来两个人互助、互补、互黑，他幽默的性格让我枯燥的学术生涯也充满了快乐，未来大家一起加油，为更幸福的未来奋斗。

最后，我要感谢我的父母，感谢这么多年来照顾和养育之恩。每次回家看到父母更加苍老了，甚是心痛。父母是唯一为你默默付出却不求回报的，希望父母能够身体健康、永葆青春、万事如意！