|  |  |
| --- | --- |
|  | 学校代码： 10246 |
|  | 学 号： 15210720124 |
|  |  |



|  |
| --- |
| 硕 士 学 位 论 文 |

（学术学位）

**基于WPF，Bootstrap和SpringMVC的大型预制构件预应力采集管理系统研发**

**Research on Prestress Acquisition and Management System of Large Pre-fabricated Component Based on WPF，Bootstrap and SpringMVC**

|  |  |
| --- | --- |
| 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**

# 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RFID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选择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，本文中还提出了一种新的回归算法，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, 回归算法

**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 RFID 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 regression 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, regression algorithm

**CLC:** TN91

# 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。通过这种方法，就可以做到长期有效的监控。

## 1.2研究前景及论文的意义

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。预应力技术是通过对钢筋施加拉应力，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，在大型预制构建生产中直接置入，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】=【7】的【1】，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模式导致了应用程序的不同方面如输入逻辑、业务逻辑和UI逻辑的分离，同时提供了在这些元素之间的松散耦合，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，主要工作为预应力信息的采集；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论文主要工作及组织结构

### 1.3.1 论文的研究内容

本文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））是现在在工程和经历等领域经常用到的一种预测模型算法，所谓灰度模型即有总舵不确定的因素，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，改变了初始值的选取和背景值的计算公式。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 论文的内容结构

本文共分为七个章节，内容分别如下：

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。前端（browser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）算法模型的缺点与不足，并提出了一种优化算法，使得对预应力的预测分析更加准确。

第六章对全文进行了总结，并结合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=13的1】。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=13的2】。

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=13的4】，有光电式无源传感器【17=13的6】，有压电式无源传感器【18=13的7】，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

预应力钢绞线是有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](http://baike.baidu.com/view/84909.htm)F，这个拉力除以杆的[截面积](http://baike.baidu.com/view/5347593.htm)S，称为“线应力”，杆的伸长量除以原长L，称为“线应变”。线应力除以线应变就等于杨氏模量，即：

 (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 | |
| 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，所以我们可以把当作是一个已知的常数；同理，和也是已知常数，根据公式(2-1)我们可以得到：

 (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大小。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时不可行的，我们需要一个装换装置间接测量钢绞线预应力的大小。该装置将在2.1.3节介绍。

### 2.1.2 应变片与应变电路

电阻应变片一般采用半导体材料利用其应变效应制作而成。应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，长度为L，横截面积为S，那么电阻值为：

 (2-3)

当金属丝在轴向力作用下：长度延伸、面积缩小、电阻率变化时，则电阻的阻值也发生相应变化。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【19=小论文的2】。但是由应变片形变造成的电阻值变化十分微小无法直接测量，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

a

b

c

d

R1

R2

R3

R4

Ei

Eo

其中R1是应变片，R2，R3，R4是普通电阻，，当应变片不受力时，R1=R2=R3=R4=R。此时b、d两点的电位都是，，输出电压为0。当应变片在外力的作用下发生形变，阻值变化为，R1阻值变为，此时b点电位发生变化，不再为0。输出电压与电阻值变化的关系为：

 (2-4)

灵敏度是应变材料的一个重要参数，他的定义为电阻的变化率与应变的比值：

 (2-5)

其中，即应变。将公式(2-5)代入公式(2-4)可以得到：

 (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、和电阻率等发生变化，最终导致阻值发生变化。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【20=小论文3】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Ra、Rb和Rt是调零电路Ra=Rb=R1=R2。当全桥应变片投入工作时，虽然温度的变化会使应变片阻值发生变化，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[21=小论文4]:

 (2-7)

图2-4 带调零电路的全桥电路

Ei

Eo

Ra

Rb

Rt

R1

R2

R3

R4

除了可以温度补偿，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装换装置来间接测量计算钢绞线预应力。如图2-5：

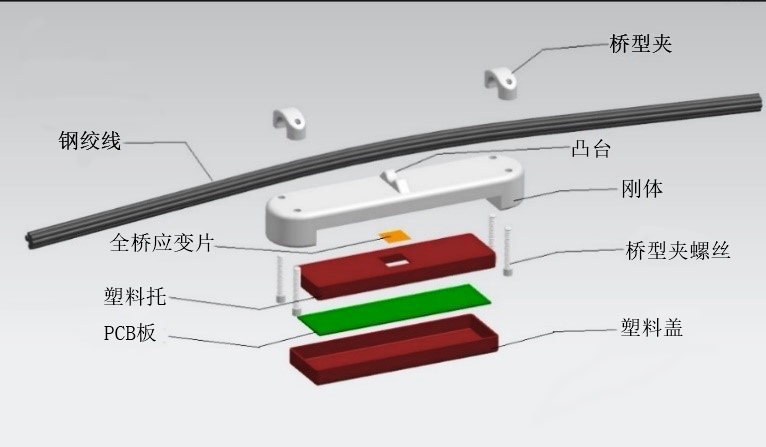


图2-5 预应力转换装置

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。可以将钢绞线的受力模型简化为图2-6所示：

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作用下发生微小形变，形变角趋向于0。根据经典力学知识我们可以知道，当预应力为F时，，合力为：







 (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。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：

 (2-9)

 (2-10)

梁的横截面某处的线应力大小【22=小论文5】为：

 (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，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：

 (2-12)

由于和a都是常数，公式(2-12)可以简化为：

 (2-13)

而应力和应变的关系为：

 (2-14)

E即杨氏模量，为刚体在x处的应变，即：

 (2-15)

故总应变为：

 (2-16)

将公式(2-8)、(2-13)代入，得到：

 (2-17)

其中为比例系数。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。

 (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。主要包括传感器的基本介绍，选择无线无源传感器的原因以及基于RFID传感器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社会和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以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：

⑴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。

⑵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。

⑶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分析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。并通过回归预测，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需求。设计了基于C/S和B/S混合结构的系统架构，如下图所示：

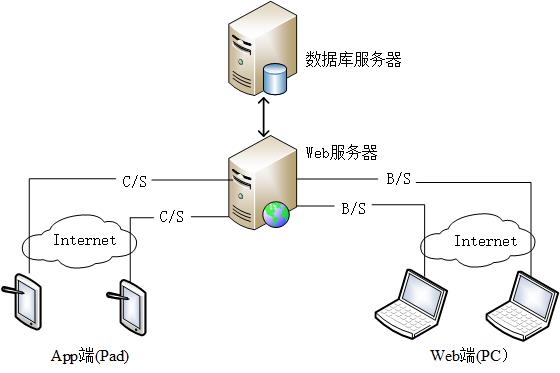


图3-1 大型预制构件采集管理系统架构图

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【23】，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。

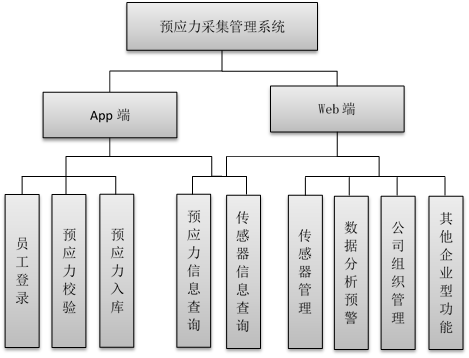


图3-2 大型预制构件采集管理系统功能图

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：

⑴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

⑵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是一个响应式页面，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。

⑶Web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Web服务器有三种：Apache、Nginx和IIS【24】。其中IIS是微软公司开发主推的服务器，由于只支持Windows环境，所以市场占有率不如Apache。Apache作为世界第一的Web服务器，市场占有率依然达到60%，它几乎可以在所有的Unix、Linux以及Windows平台运行。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。

⑷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【RSA AND MD5】。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【基于RSA】。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：

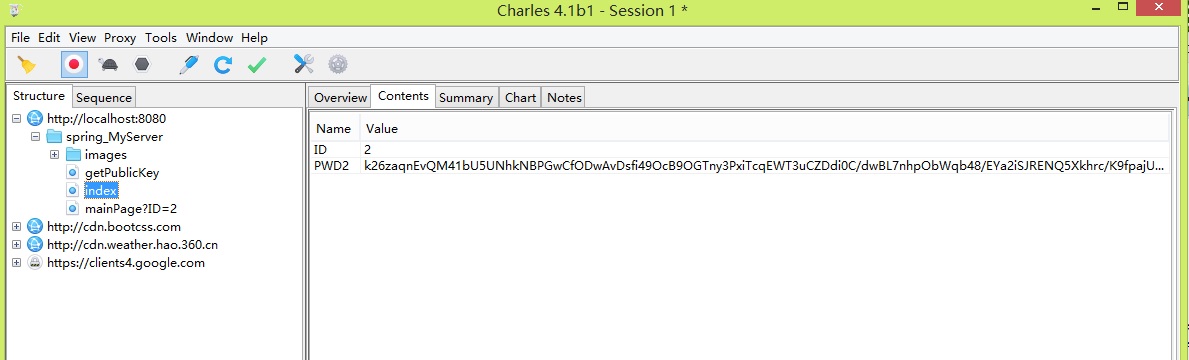


图3-4 登陆验证流程图

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【25=小论5】。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 Framework4.5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。

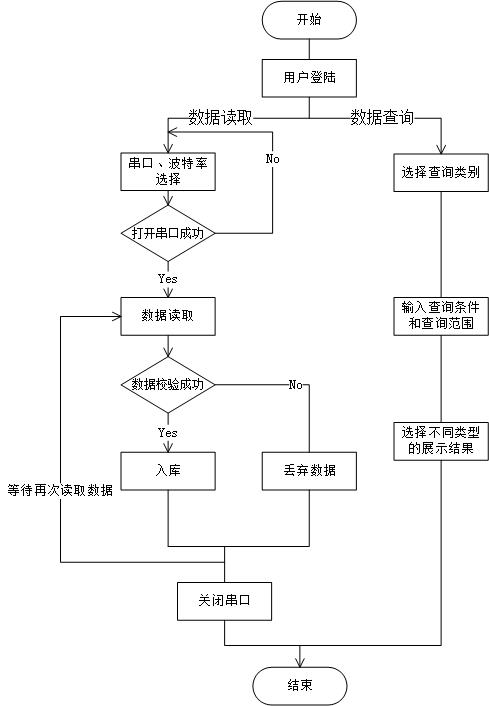


图4-1 App端工作流程图

### 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框架采取得是MVVM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）表达相对应的数据。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数据生动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线图等，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【25】。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【26】。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：

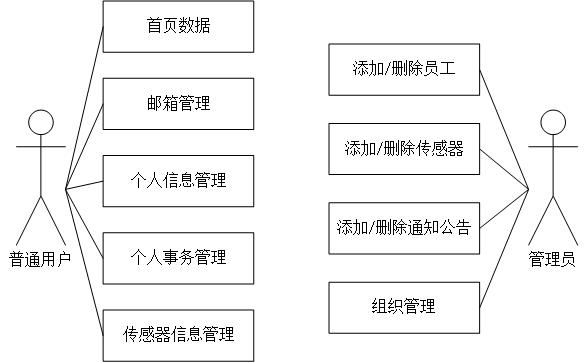


图4-7 网站用例图

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。用小人表示参与者，用椭圆表示用例。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表率的显示器。

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【27】。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**

为了使人机交互更加友好，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jQlouds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【28】,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）【29】。

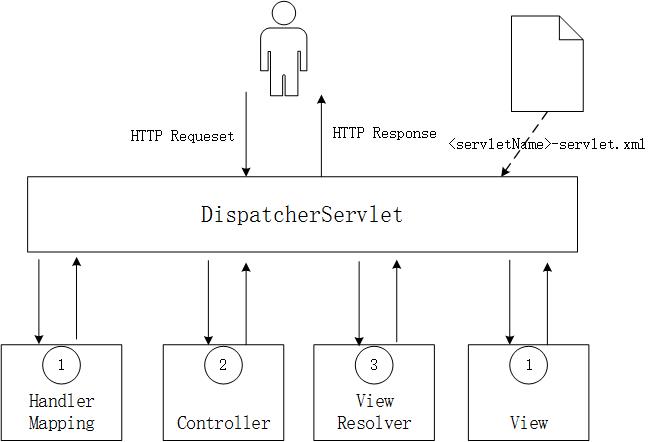


图4-13 SpringMVC工作流程图

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编码

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。通常负责数据库的相应数据存取。

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）负责从视图读取数据，控制用户输入，并向模型发送数据。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

数据库的是一个系统必不可少的存储设备。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【30】。根据数据库设计的三大基础原则【31的5】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用户的基本信息。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。

1. 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\_radio | 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。

1. 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【32】。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【33】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【34】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【35】。

## 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，如果满足级比：

 (5-1)

在区间内，就可以使用GM（1,1）模型建模。由于序列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：

 (5-2)

由生成背景序列，其中

 (5-3)

一般取0.5。

在文献[36]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 不同年份不同节点预应力值

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

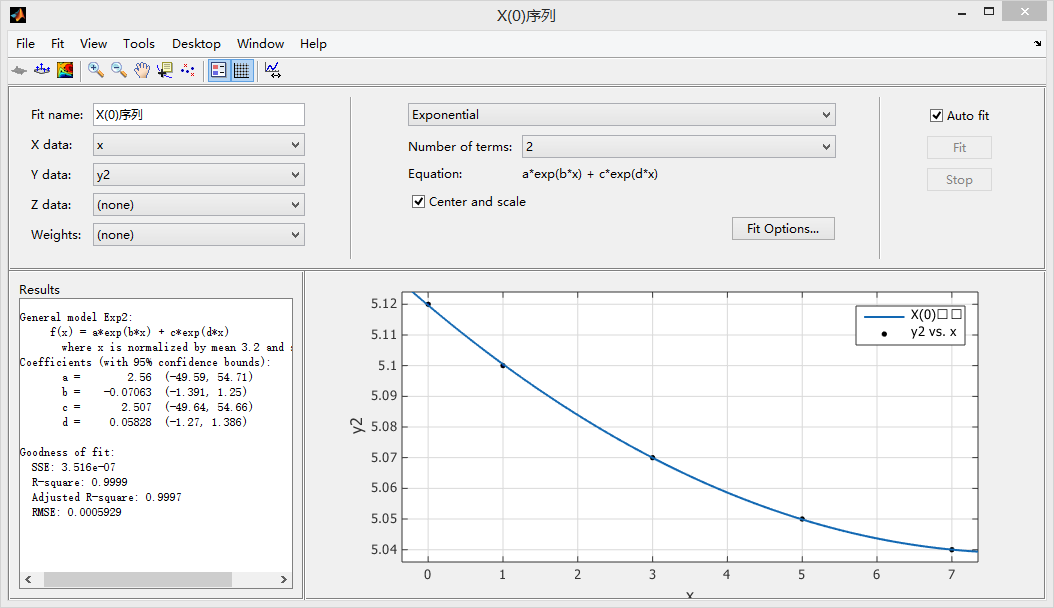


图5-2 序列拟合图

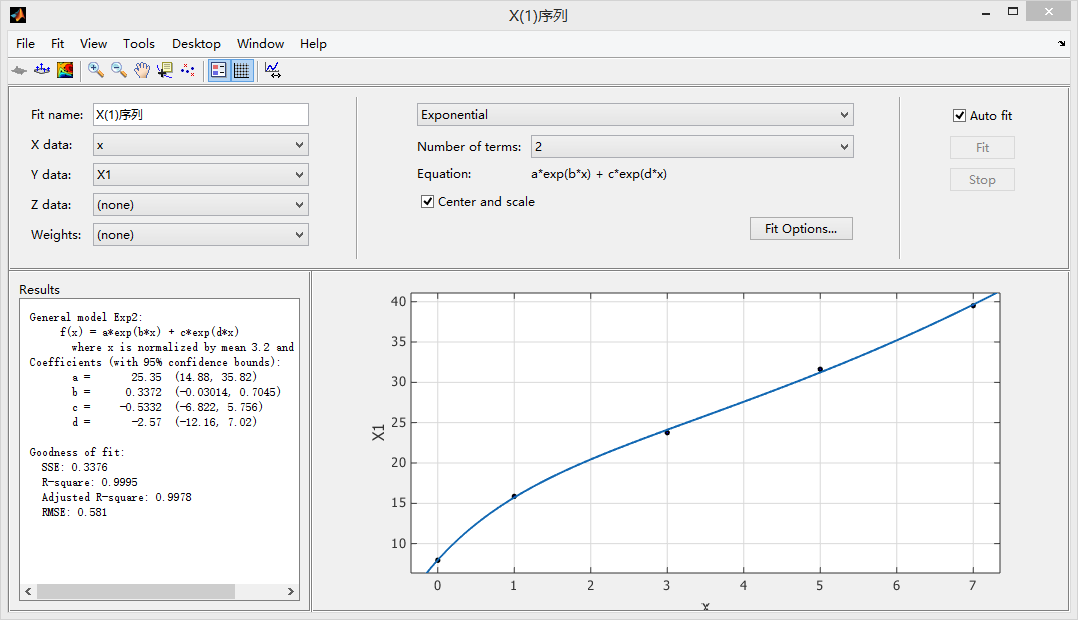


图5-3 序列拟合图

由图可知不论是还是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：

 (5-4)

在离散序列中，我们可以将微分方程变成差分：

 (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：，，，可以用最小二乘法求得:

 （5-6）

而和的预测公式分别如下：

 （5-7）

 （5-8）

将公式（5-6）中得到的a，b的预测值代入就可以得到和的具体值。

## 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以及初始值的选择有重要关系。因为背景序列表示的是在[*k,k+*1]这段时间内的背景值，若在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，初始值的选择也不是最优，因为它是一个非累计值，具有随机性【37】,我们之所以选择而不是就是希望从累加值发现变化规律，降低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【38=37的4】中已经提出了新的预测公式：

 (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作为初始数据，我们在这里借用此公式。此外传统的公式：，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。在此处提出了一种新的背景值就算公式：

 (5-10)

通过改变和前的系数，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：

 (5-11)

另k=4，即可算出为25.3654，故=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：

 (5-12)

另k=4，即可算出为25.3687，故=5.0287。

对比原值5.04，可知改进型GM（1,1）模型预测更准确。

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

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，本文创新地提出使用物联网的技术，实现对传统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. 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，可以将预应力数据装换为电信号。
2. 设计了预应力信息采集管理系统。

# 参考文献

# 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和计算》，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，也是在复旦，我认识了我女朋友施颖培，感谢她这三年的陪伴和激励，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，还是要在此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，感谢这么多年来的照顾和养育之恩。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！