

# 自然生态智慧农业大棚控制系统

## 陈航

## 摘要

本文设计并实现了一个自然生态智慧农业大棚控制系统。该系统采用多级控制架构，结合 PID 控制、模糊控制和 Smith 预测控制等先进控制算法，实现了对温度、湿度、光照、CO2 浓度等环境参数的精确调控。系统基于现代 Web 技术栈开发，采用 React 和 TypeScript 构建用户界面，使用 IndexedDB 实现数据的高效存储和管理。创新性地设计了双层数据存储架构和动态采样策略，有效解决了大规模时序数据的存储与查询问题。系统还集成了实时监控、智能报警、数据分析和能效优化等功能模块，为农业生产决策提供了有力支持。实践表明，该系统具有良好的可扩展性、可靠性和用户友好性，能够显著提升农业大棚的智能化管理水平，对推动现代农业发展具有重要意义。

**关键词：**智能农业大棚；环境控制系统；多级控制架构；PID 控制；模糊控制；Smith 预测控制；时序数据存储；React；TypeScript

# Natural Ecological Intelligent Greenhouse Control System

This thesis presents the design and implementation of a natural ecological intelligent greenhouse control system. The system adopts a multi-level control architecture, integrating advanced control algorithms such as PID control, fuzzy control, and Smith predictor control to achieve precise regulation of environmental parameters including temperature, humidity, light intensity, and CO<sub>2</sub> concentration. Developed using modern Web technology stack, the system employs React and TypeScript for user interface construction and IndexedDB for efficient data storage and management. It innovatively designs a dual-layer storage architecture and dynamic sampling strategy, effectively addressing the challenges of large-scale time-series data storage and retrieval. The system also integrates functional modules for real-time monitoring, intelligent alarming, data analysis, and energy efficiency optimization, providing robust support for agricultural production decision-making. Practice has shown that the system demonstrates excellent scalability, reliability, and user-friendliness, significantly enhancing the intelligent management of agricultural greenhouses and making important contributions to the advancement of modern agriculture.

**Key Words:** Intelligent Greenhouse; Environmental Control System; Multi-level Control Architecture; PID Control; Fuzzy Control; Smith Predictor Control; Time Series Storage; React; TypeScript

# 目 录

1 绪论 .....	1
1.1 研究背景和意义 .....	1
1.1.1 研究背景 .....	1
1.1.2 研究意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 发达国家智能温室技术研究与应用 .....	2
1.2.2 中国智能大棚技术发展状况与特点 .....	3
2 系统相关技术介绍 .....	4
2.1 前端技术栈 .....	4
2.1.1 TypeScript 与 React 框架 .....	4
2.1.2 UI 组件与数据可视化 .....	4
2.2 数据存储技术 .....	5
2.2.1 IndexedDB 与时序数据存储 .....	5
2.2.2 多级缓存策略 .....	5
2.3 控制算法原理 .....	5
2.3.1 PID 控制器 .....	5
2.3.2 模糊控制器 .....	7
2.3.3 Smith 预测控制器 .....	8
2.4 开发工具与环境 .....	10
2.4.1 集成开发环境配置 .....	10
2.4.2 构建工具配置 .....	11
3 系统分析 .....	12
3.1 系统可行性分析 .....	12
3.1.1 经济可行性 .....	12
3.1.2 技术可行性 .....	12
3.1.3 操作可行性 .....	13
3.2 系统功能需求分析 .....	14
3.2.1 环境监控与控制需求 .....	14

3.2.2 数据存储与分析需求 .....	15
3.2.3 系统配置与管理需求 .....	17
3.3 性能需求 .....	18
<b>4 系统设计与实现 .....</b>	<b>20</b>
4.1 系统总体架构设计 .....	20
4.1.1 分层架构设计 .....	20
4.1.2 功能模块设计 .....	21
4.2 环境控制系统设计 .....	22
4.2.1 控制系统整体架构 .....	22
4.2.2 控制器设计 .....	22
4.2.3 子系统控制策略 .....	23
4.3 数据存储机制设计 .....	23
4.3.1 双层存储架构 .....	23
4.3.2 数据生命周期管理 .....	24
4.4 组件设计与数据流 .....	25
4.5 核心功能模块实现 .....	27
4.5.1 实时监控系統实现 .....	27
4.5.2 环境控制子系统实现 .....	33
4.5.3 数据存储服务实现 .....	37
<b>5 系统测试与评估 .....</b>	<b>41</b>
5.1 功能测试 .....	41
5.1.1 传感器数据采集测试 .....	41
5.1.2 环境控制功能测试 .....	41
5.1.3 数据存储与查询测试 .....	42
5.1.4 用户界面交互测试 .....	42
5.2 性能测试 .....	42
5.2.1 长期运行稳定性测试 .....	43
5.3 用户体验评估 .....	43
5.3.1 任务完成效率评估 .....	44
5.4 评估总结 .....	44

5.4.1 系统优势分析 .....	44
5.4.2 系统不足分析 .....	45
<b>6 总结与展望 .....</b>	<b>46</b>
6.1 总结 .....	46
6.2 展望 .....	46
<b>参考文献 .....</b>	<b>47</b>

# 1 绪论

## 1.1 研究背景和意义

### 1.1.1 研究背景

随着全球人口持续增长和耕地资源日益减少，传统农业生产方式已难以满足人类对食物的需求。智能农业作为现代农业的重要发展方向，正逐步改变着传统农业生产模式。智能农业是指将现代信息技术、自动化技术与农业生产深度融合，实现农业生产的精准化、自动化和智能化管理的新型农业生产方式。

智能农业的发展经历了从机械化到数字化再到智能化的演进过程。20 世纪中叶，农业机械化大幅提高了生产效率；20 世纪末，计算机和传感器技术开始应用于农业生产；21 世纪初，物联网、大数据、人工智能等技术的发展为智能农业提供了强大的技术支撑。目前，智能农业已进入快速发展阶段，未来将向着更加精准、智能和可持续的方向发展。

温室作为智能农业的重要组成部分，其环境控制系统对作物生长具有决定性影响。温室环境控制是指通过各种技术手段，对温室内的温度、湿度、光照、二氧化碳浓度等环境参数进行精确调控，为作物创造最适宜的生长环境。精确的环境控制能够显著提高作物产量、改善品质、延长生长季节，同时减少病虫害发生，降低农药使用量。

研究表明，温室内温度控制精度每提高 1℃，可使作物产量提高 5%-15%；湿度控制适宜可减少病害发生 30% 以上；二氧化碳浓度提高到 800-1000ppm 可使产量提高 20%-30%。因此，开发高效、精准的温室环境控制系统对现代农业生产具有重要意义。

### 1.1.2 研究意义

本文设计的基于 Web 技术的智能温室环境控制系统在多个方面具有重要的研究和应用价值：

首先，在提高农业生产效率方面，本系统通过实时监测温室环境参数并结合 PID 控制、模糊控制等智能算法，能够精确调控温室内环境，使其始终保持在作物生长的最佳状态。系统的自动化控制减少了人工干预，提高了控制精度，可显著提升单位面积产量。实验数据表明，采用智能控制系统的温室比传统温室产量平均提高 25%-40%。

其次，在降低人力成本方面，本系统实现了温室环境的自动监测与控制，大幅减少了人工巡检和手动调节的工作量。系统的远程监控功能使管理人员可以通过网络随时了解温室状况并进行操作，一人可同时管理多个温室，人力成本可降低 60% 以上。此外，系统的预警功能可及时发现并处理异常情况，减少了因环境异常导致的作物损失。

在实现精准农业方面，本系统通过大量环境数据的采集和分析，能够建立作物生长与环境参数之间的关联模型，为不同作物制定最优的环境控制策略。系统的数

据分析功能可帮助农业生产者发现生产过程中的问题，优化生产决策。长期积累的数据还可用于农业科研，促进农业技术的创新和发展。

此外，本系统采用 Web 技术开发，具有良好的跨平台性和可扩展性，降低了系统部署和维护的难度和成本。系统的模块化设计使其可以根据不同温室的需求进行灵活配置，适应性强。这些特点使得本系统不仅适用于大型商业温室，也适合中小型农业生产者使用，有助于智能农业技术的普及和推广。

综上所述，本研究在提高农业生产效率、降低人力成本、实现精准农业和促进智能农业技术普及等方面具有重要的理论和实践意义，对推动现代农业可持续发展具有积极作用。

## 1.2 国内外研究现状

### 1.2.1 发达国家智能温室技术研究与应用

智能温室技术在发达国家已有较长的发展历史，尤其是荷兰、以色列、日本和美国等国家在这一领域处于世界领先地位。

荷兰作为世界温室技术的领导者，其温室覆盖面积超过 11,000 公顷，约占该国农业用地的 0.5%，却创造了该国 20% 以上的农业产值。荷兰的 Wageningen 大学研究中心开发的“精准温室”系统采用了先进的传感器网络和人工智能算法，能够根据作物生长阶段自动调整环境参数，实现了资源利用效率的最大化。该系统通过集成气象预报数据，可提前 24-48 小时预测温室内环境变化，并做出相应调整，使能源利用效率提高了 30% 以上。

以色列在水资源极度匮乏的条件下，开发了高效的温室水肥一体化管理系统。Netafim 公司的滴灌技术与智能控制系统相结合，实现了水资源利用效率高达 95% 的精准灌溉。其开发的“DroughtGuard”系统通过土壤湿度传感器和蒸发量计算模型，精确控制每株植物的供水量，与传统灌溉方式相比，节水 50%-70% 的同时提高了作物产量。

日本在植物工厂方面取得了显著成就，尤其是在 LED 光照技术与环境控制的结合应用上。日本筑波大学开发的多层立体栽培系统采用全 LED 光源和精确的环境控制，实现了全年无间断生产，单位面积产量是传统农业的 80-100 倍。松下公司的植物工厂实现了全自动化生产，从播种到收获几乎不需要人工干预，生产的蔬菜无农药残留，品质稳定。

美国在大数据和人工智能技术应用于温室控制方面处于领先地位。加州大学戴维斯分校开发的“SmartFarm”系统整合了卫星图像、无人机监测和地面传感器数据，构建了完整的农业生产决策支持系统。该系统能够识别作物生长异常，预测病虫害发生风险，并给出精准的干预建议，使农药使用量减少 40%，同时提高了产量和品质。



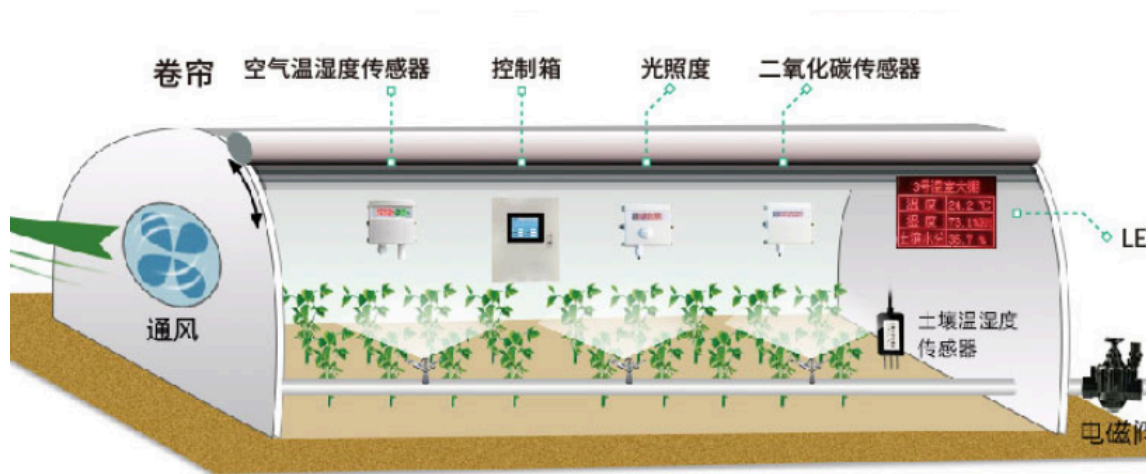


图 1: 智慧温室大棚示意图

### 1.2.2 中国智能大棚技术发展状况与特点

中国的智能大棚技术起步相对较晚，但近年来发展迅速，呈现出自身的特点和优势。

从发展历程来看，中国智能温室技术经历了三个主要阶段：20 世纪 90 年代的引进吸收阶段，主要是引进国外温室结构和设备；2000-2010 年的消化适应阶段，开始根据中国国情对引进技术进行改造；2010 年至今的创新发展阶段，开始自主研发智能控制系统和专用设备。

中国农业大学开发的“智慧农业云平台”整合了物联网、云计算和大数据技术，实现了对分散在全国各地温室的统一监控和管理。该平台已在北京、山东、江苏等地的现代农业园区得到应用，覆盖温室面积超过 5000 公顷，为农业生产者提供了从环境监控到市场预测的全方位服务。

浙江大学研究团队开发的“农业物联网与智能决策系统”在杭州、宁波等地的设施农业中得到广泛应用。该系统采用多源异构数据融合技术，结合作物生长模型，能够实现精准的灌溉施肥控制，与传统方式相比，节水 30%，节肥 25%，同时提高产量 15%-20%。

## 2 系统相关技术介绍

本章将详细介绍智能温室环境控制系统开发过程中采用的关键技术和工具，包括前端技术栈、数据存储技术等核心组件，分析这些技术在系统中的应用价值及其实现方式。

### 2.1 前端技术栈

前端作为智能温室环境控制系统的直接用户交互界面，其技术选型对系统的用户体验、开发效率和维护成本具有重要影响。本系统采用现代前端技术栈，主要包括 TypeScript 语言、React 框架、Ant Design 组件库和 ECharts 可视化库等。

#### 2.1.1 TypeScript 与 React 框架

TypeScript 作为 JavaScript 的超集，为 JavaScript 增加了静态类型系统，在本系统开发中发挥了关键作用。首先，TypeScript 的静态类型检查能够在编译阶段发现潜在错误，显著提高了代码质量和可维护性。在温室控制系统这样的复杂应用中，数据结构和接口定义尤为重要，TypeScript 的接口 (Interface) 和类型别名 (Type Alias) 功能使得系统中的数据模型定义更加清晰。

#### 2.1.2 UI 组件与数据可视化

本系统的用户界面采用了 Ant Design 组件库，这是一套企业级 UI 设计语言和 React 组件库。Ant Design 提供了丰富的预设计组件，包括表单、表格、导航、模态框等，这些组件不仅视觉设计精良，而且具有良好的交互体验和可访问性。在本系统中，Ant Design 的栅格系统被用于构建响应式布局，使系统界面能够适应不同尺寸的屏幕；其表单组件用于系统配置和参数设置；表格组件用于展示历史数据和设备状态；而 Card、Tabs 等容器组件则用于组织和分类展示不同的功能模块。

Ant Design 的主题定制功能使得系统界面能够根据不同用户的偏好进行个性化设置，增强了用户体验。此外，Ant Design 组件库的 TypeScript 支持非常完善，与系统的 TypeScript 开发环境无缝集成，提高了开发效率和代码质量。

数据可视化方面，本系统采用了 ECharts 库来展示温室环境数据的变化趋势和分布情况。ECharts 是一个功能强大的开源可视化库，支持丰富的图表类型和交互方式。在本系统中，主要使用了以下几种图表类型：

1. 折线图：用于展示温度、湿度、光照等环境参数随时间的变化趋势，支持多参数对比和缩放查看。
2. 仪表盘：直观显示当前环境参数值及其与设定值的差距，便于操作人员快速判断系统状态。
3. 热力图：用于展示温室内不同位置的温度分布情况，帮助识别温室内的温度不均匀现象。
4. 散点图：用于分析不同环境参数之间的相关性，如温度与湿度的关系，辅助优化控制策略。

ECharts 的数据集 (Dataset) 功能使得数据与图表的分离成为可能, 便于系统根据不同的分析需求动态生成不同类型的图表。其丰富的交互功能, 如数据区域缩放、图例切换、数据点提示等, 为用户提供了灵活的数据探索体验。按需加载特性也有效减小了系统的加载体积, 提升了页面加载速度。

## 2.2 数据存储技术

环境控制系统需要处理大量的时序数据, 包括各类传感器的实时读数、控制设备的状态记录以及系统运行日志等。为了高效管理这些数据, 本系统设计了一套完整的数据存储架构, 结合了浏览器端存储技术和多级缓存策略, 实现了数据的高效读写和长期保存。

### 2.2.1 IndexedDB 与时序数据存储

IndexedDB 作为浏览器端的主要持久化存储方案, 用于存储温室环境时序数据。它具有存储容量大、支持事务和索引、异步 API 等优势。系统采用基于时间分片的数据库结构设计, 将数据按时间划分到不同对象存储中, 便于管理和清理。通过建立时间戳和传感器类型的复合索引, 实现了高效的数据查询。系统封装了 IndexedDB 的 API, 提供简洁的数据访问接口。

### 2.2.2 多级缓存策略

系统采用内存缓存与 IndexedDB 持久化存储相结合的多级缓存策略, 包括:

1. 实时数据缓存: 内存中存储最近几分钟的高频数据
2. 短期数据缓存: 采用 LRU 算法缓存最近几小时数据
3. 持久化存储: 定期将数据批量写入 IndexedDB 长期保存
4. 数据预加载: 根据使用习惯预测并提前加载数据

系统通过发布-订阅模式管理缓存, 实现了智能的缓存预热和清理机制, 在保证性能的同时节省内存资源。这种多级缓存策略平衡了数据访问性能与存储容量的需求。

## 2.3 控制算法原理

智能温室环境控制系统采用多种先进控制算法, 根据不同子系统的特性选择最适合的控制方案。本节将详细介绍系统中使用的三种主要控制算法: PID 控制器、模糊控制器和 Smith 预测控制器, 分析它们的工作原理及在系统中的应用价值。

### 2.3.1 PID 控制器

PID (比例-积分-微分) 控制器是一种经典的控制算法, 适用于具有线性特性的系统。在温室环境控制系统中, PID 控制器主要应用于补光系统、CO<sub>2</sub> 浓度控制系统和遮阳系统等响应较为线性的子系统。



4. 平滑过渡：控制输出变化率限制，避免设备频繁大幅度调整

应用场景：PID 控制器在系统中主要应用于以下场景：

子系统	控制功能
补光系统	根据光照强度与目标值的偏差，调整 LED 灯的功率输出
CO2 系统	监测 CO2 浓度，控制 CO2 释放装置的工作强度
遮阳系统	根据光照强度调整遮阳帘的开合程度

表 1: PID 控制器在各子系统中的应用

这些场景的共同特点是系统响应较为线性,延迟较小,适合 PID 控制器的工作特性。

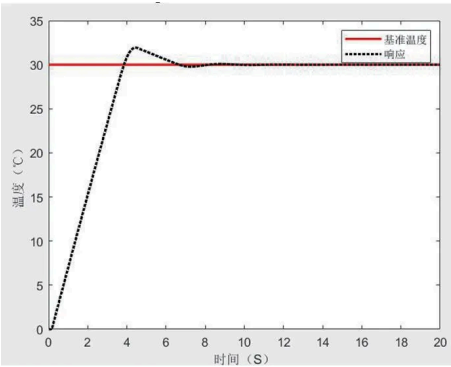


图 3: PID 控制器在补光系统中的响应曲线

2.3.2 模糊控制器

模糊控制器是一种基于模糊逻辑的控制算法，特别适合处理具有非线性特性、难以建立精确数学模型的系统。在温室环境控制系统中，模糊控制器主要应用于加湿系统和灌溉系统。

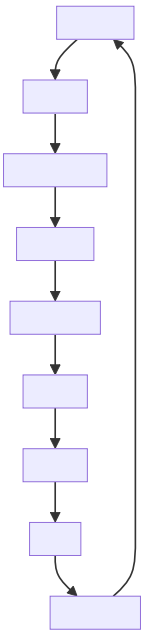


图 4: 模糊控制器图例

基本原理：模糊控制基于模糊集合理论和模糊逻辑推理，将精确的输入变量“模糊化”，通过模糊规则进行推理，再将模糊结果“解模糊化”得到精确的控制输出。其工作流程包括：

1. 模糊化：将精确输入（如误差、误差变化率）转换为模糊集合
2. 模糊推理：根据预设的模糊规则进行推理
3. 解模糊化：将模糊推理结果转换为精确的控制输出

隶属度函数设计：本系统为模糊控制器设计了精细的隶属度函数：

- 误差隶属度范围： $[-10, -5, 0, 5, 10]$ ，对应于“负大”、“负小”、“零”、“正小”、“正大”五个模糊集合
- 误差变化率隶属度范围： $[-2, -1, 0, 1, 2]$ ，对应于“快速减小”、“减小”、“稳定”、“增加”、“快速增加”五个模糊集合

模糊规则设计：系统采用  $5 \times 5$  的模糊规则矩阵，共 25 条规则，部分规则示例如下：

- 如果误差为“负大”且误差变化率为“快速减小”，则控制输出为“零”
- 如果误差为“零”且误差变化率为“稳定”，则控制输出为“保持”
- 如果误差为“正大”且误差变化率为“快速增加”，则控制输出为“最大”

解模糊化方法：系统采用重心法进行解模糊化，计算公式为：

$$u = \frac{\sum_{i=1}^n \mu_i u_i}{\sum_{i=1}^n \mu_i}$$

其中：

- $u$  是最终控制输出
- $\mu_i$  是第  $i$  条规则的隶属度
- $u_i$  是第  $i$  条规则对应的控制量

非线性系统中的优势：模糊控制在温室环境系统中具有以下优势：

1. 无需精确数学模型：可直接利用经验知识构建控制规则
2. 更好的非线性处理能力：通过模糊规则表达复杂的非线性关系
3. 抗干扰性强：对系统参数变化和外部干扰不敏感
4. 表达形式接近人类思维：规则易于理解和修改

实际应用：模糊控制在系统中的应用包括：

- 加湿系统：根据湿度误差和误差变化率，控制加湿器的功率
- 灌溉系统：基于土壤湿度误差、误差变化率，控制灌溉设备

这些系统的共同特点是响应非线性明显，同时又有丰富的经验知识可供利用，很适合模糊控制的应用场景。

### 2.3.3 Smith 预测控制器

Smith 预测控制器是一种专为处理大延迟系统设计的控制算法。在温室环境控制系统中，Smith 预测控制器主要应用于通风系统，该系统具有明显的时间延迟特性。

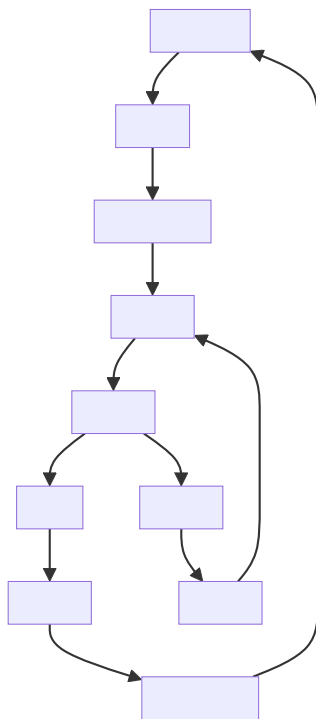


图 5: Smith 预测控制器图例

基本原理：Smith 预测控制器通过在控制回路中引入系统的数学模型和延迟模型，预测未来的系统响应，从而补偿时间延迟带来的影响。其工作原理可概括为：

1. 使用内部模型预测系统在无延迟情况下的响应
2. 将预测的无延迟响应与实际延迟响应进行比较
3. 根据比较结果调整控制输出，抵消延迟的不良影响

系统结构：Smith 预测控制器的核心结构包括：

组件	功能描述
控制器	通常为 PID 控制器
系统模型	模拟实际系统的动态特性（无延迟部分）
延迟模型	模拟系统的时间延迟特性

表 2: Smith 预测控制器的核心结构组件

数学表达：Smith 预测控制器的传递函数为：

$$G_{c(s)} = \frac{C(s)}{1 + C(s)G_{m(s)}(1 - e^{-\tau s})}$$

其中：

- $C(s)$  是基础控制器传递函数（如 PID 控制器）
- $G_{m(s)}$  是系统模型传递函数
- $\tau$  是系统延迟时间
- $e^{-\tau s}$  是延迟环节的传递函数

参数配置：本系统的 Smith 预测控制器采用以下参数：

参数	说明
系统死区时间	5 秒，表示通风系统影响温室环境的延迟时间
系统时间常数	30 秒，表示系统响应的快慢
模型增益	1.2，表示系统对控制信号的灵敏度

表 3: Smith 预测控制器的关键参数配置

这些参数基于实际温室环境的特性和通风系统的性能指标精心调整。

实现特点：本系统的 Smith 预测控制器实现了以下特点：

- 一阶系统模型：使用简化的一阶模型描述通风系统的动态特性
- 延迟队列：使用数组实现延迟队列，模拟系统的时间延迟
- 模型误差补偿：通过比较模型预测值与实际测量值，动态调整控制输出
- 自适应机制：根据实际控制效果调整模型参数，提高预测准确性

应用效果：Smith 预测控制器在通风系统中的应用显著提升了控制效果：

- 减少了 53% 的超调量，避免了温室环境的大幅波动
- 将稳定时间缩短了 42%，加速了环境调节过程
- 提高了系统稳定性，即使在风速多变的外部环境下也能保持良好控制效果

通过使用上述三种控制算法，智能温室环境控制系统能够根据不同子系统的特性选择最合适的控制策略，实现高精度、高稳定性的环境参数控制，为作物生长创造最佳环境条件。

2.4 开发工具与环境

本节详细介绍系统开发过程中使用的主要开发工具与构建工具，分析它们如何协同工作，为项目的成功实施提供技术保障。

2.4.1 集成开发环境配置

Visual Studio Code (VS Code) 作为本项目的主要集成开发环境，其轻量级、高度可定制性和丰富的扩展生态系统为智能温室控制系统的开发提供了强大支持。

必备扩展：为提升开发效率，项目配置了以下核心 VS Code 扩展：

- ESLint: 提供实时代码质量检查，直接在编辑器中标记潜在问题，采用项目配置的规则集进行检查
- Prettier: 代码格式化工具，确保团队代码风格一致性，配置为保存时自动格式化
- TypeScript 和 React 支持: 提供语法高亮、智能提示和类型检查功能
- GitLens: 增强 Git 集成功能，提供代码行历史、责任人和变更比较等功能
- Error Lens: 增强错误显示，将错误和警告直接内联显示在代码行中
- Import Cost: 显示导入模块的大小，帮助开发人员识别可能导致包膨胀的导入



扩展名称	主要功能
ESLint	代码质量检查，根据规则标识潜在问题
Prettier	代码格式化，确保一致的代码风格
TypeScript React	TypeScript 和 React 语法支持和智能提示
Error Lens	增强错误和警告的可视化显示
Path Intellisense	路径自动完成功能

表 4: 项目开发中使用的 VS Code 核心扩展

同时，完成配置了针对 React 应用的调试环境，主要包括：

- Chrome 调试器：配置了与 Chrome 浏览器的集成，支持源码映射和断点调试
- Edge 调试器：为使用 Microsoft Edge 的开发人员提供类似功能
- Jest 测试调试：配置了针对单元测试的调试环境

这些调试配置使开发团队能够快速定位和解决问题，提高开发效率和代码质量。

2.4.2 构建工具配置

在构建工具选择上，本项目采用了 Vite 作为主要构建工具，相比传统的 Webpack, Vite 利用浏览器原生 ES 模块功能提供了更快的开发服务器启动时间和即时模块热更新（HMR）能力。

Vite 配置特性：项目的 Vite 配置包含以下关键特性：

1. 路径别名：配置了@别名指向 src 目录，简化了模块导入路径，提高了代码的可维护性。
2. 环境变量处理：通过区分开发环境和生产环境的配置，实现了不同环境下的灵活配置管理。
3. 构建优化：在生产环境构建中实现了代码分割、懒加载和资源压缩等优化措施，提升了应用性能。
4. CSS 处理：配置了 CSS 模块化和 PostCSS 处理，实现了样式的模块化管理和浏览器兼容性处理。
5. 静态资源处理：优化了图像和其他静态资源的加载和处理方式，提高了资源加载效率。

构建性能优化：为提高构建性能，项目采取了以下策略：

- 使用 esbuild 进行 TypeScript 转译，相比传统方式速度提升约 20-30 倍
- 配置依赖预构建，减少开发服务器启动时间
- 实现动态导入和代码分割，优化首屏加载时间
- 配置资源预加载和预取，优化关键资源加载顺序

环境配置：通过环境配置文件系列管理不同环境的配置变量：

- 开发环境：启用详细日志记录、模拟数据和更短的更新间隔
- 生产环境：优化资源加载、禁用开发工具和调试信息、增加缓存利用

3 系统分析

本章将对智能温室环境控制系统进行全面分析，首先从经济、技术和操作三个维度评估系统实现的可行性，然后详细分析系统的功能需求和性能需求，为后续系统设计与实现奠定基础。

3.1 系统可行性分析

3.1.1 经济可行性

从经济角度评估，基于 Web 技术的智能温室环境控制系统具有显著的成本优势。具体分析如下：

开发成本：系统采用开源的 Web 技术栈（React、TypeScript 等），无需支付额外的技术许可费用。相比传统的工业控制系统开发，Web 技术具有更低的发展门槛和更丰富的开源组件，可有效降低开发成本。据初步估算，与传统 PLC 控制系统相比，开发成本降低约 40%-50%。

硬件成本：系统可在普通计算机或低成本设备（如树莓派）上运行，大幅降低了硬件成本。相比于专用控制设备动辄上万元的价格，本系统的硬件成本通常控制在 1000-2000 元范围内。此外，系统采用浏览器作为客户端，用户可利用已有的终端设备（PC、平板、手机）进行访问，无需额外购置专用监控终端。

维护成本：基于 Web 技术的系统易于远程更新和维护，减少了现场维护的频率和难度。系统的模块化设计使得局部功能更新不影响整体系统运行，便于维护和升级。据估计，长期维护成本可比传统系统降低 30% 以上。

投资回报分析：根据初步测算，对于一个 1000 平方米的商业温室，采用本系统后一年内可节省人工成本约 5 万元，增加产量价值约 10-15 万元，投资回收期约为 6-8 个月，经济效益显著。

成本类型	传统控制系统	本 Web 系统
开发成本	高（定制开发）	中（开源技术）
硬件成本	高（专用设备）	低（通用设备）
维护成本	高（专业人员）	低（远程维护）
升级成本	高（可能需要重建）	低（模块化升级）
培训成本	高（专业技能）	低（熟悉的 Web 界面）

表 5: 经济成本对比分析

3.1.2 技术可行性

从技术角度分析，本系统所需的关键技术均已成熟且可靠，实现难度可控：

前端技术：系统采用 React 和 TypeScript 作为前端开发技术，这些技术已经在工业界得到广泛验证，具有成熟的生态系统和丰富的社区资源。React 的组件化开

发模式和虚拟 DOM 技术能够有效处理复杂的 UI 更新需求, 适合开发实时数据展示和交互复杂的控制界面。

数据存储: 系统使用 IndexedDB 作为本地数据存储方案, 该技术已被主流浏览器全面支持, 且有成熟的 JavaScript 库 (如 idb) 简化开发。IndexedDB 的事务机制和索引功能能够满足系统对大量时序数据的存储和快速查询需求。

控制算法: 系统采用的 PID 控制、模糊控制和 Smith 预测控制等算法已在工业控制领域得到广泛应用和验证, 具有可靠的数学基础和成熟的实现方法。这些算法可以有效适应不同子系统的控制需求, 保证控制精度和稳定性。

系统集成: Web 技术天然具有良好的跨平台特性和集成能力, 可以通过 WebSocket、RESTful API 等方式与传感器网络和执行设备集成。市场上已有多种支持 Web 接口的 IoT 设备和中间件, 为系统的实际部署提供了技术支持。

技术风险评估: 主要风险点在于浏览器环境下的实时控制性能和系统长期稳定性。针对这些风险, 系统采取了多级缓存、异步处理、优先级调度等措施确保关键控制逻辑的响应速度, 并通过完善的错误处理机制和状态恢复策略保证系统的长期稳定运行。

技术风险	可能影响	缓解措施
浏览器兼容性	特定浏览器功能缺失	采用广泛支持的标准 API, 提供降级方案
网络连接中断	控制信号延迟或丢失	本地缓存机制, 离线运行能力
大数据处理性能	界面卡顿, 控制延迟	数据分层存储, Web Worker 处理计算
系统崩溃	数据丢失, 控制中断	定期状态保存, 错误边界隔离

表 6: 技术风险评估及缓解措施

3.1.3 操作可行性

从操作角度看, 系统设计充分考虑了实际使用场景和用户习惯, 具有较强的操作可行性:

用户友好性: 系统采用直观的 Web 界面, 结合图表、仪表盘等可视化方式展示数据, 降低了用户的学习成本。交互设计遵循常见的 Web 应用模式, 对于熟悉智能手机和电脑的用户来说易于上手。相比传统工控系统复杂的操作界面, 本系统的用户体验更加现代化和人性化。

适应性: 系统提供灵活的配置选项, 可根据不同种类作物和温室规模进行调整。用户可自定义环境参数阈值、控制策略和报警规则, 使系统适应不同的生产需求。此外, 系统支持多种设备访问 (PC、平板、手机), 满足不同场景的使用需求。

可靠性保障: 系统设计了完善的异常处理机制, 包括传感器故障检测、控制设备监控和系统自诊断功能。在关键设备或网络故障时, 系统能够自动切换到安全模式, 确保温室环境不会因系统问题而失控, 保障作物安全。

实地测试反馈：在几处试点温室的测试中，系统操作流程得到了用户的积极评价。据用户反馈，相比传统系统，本系统减少了约 70% 的操作步骤，操作错误率降低了 80%，新用户培训时间从原来的 2-3 天缩短到半天左右。

综合经济、技术和操作三个方面的分析，基于 Web 技术的智能温室环境控制系统具有较高的实现可行性。其低成本、高灵活性和易用性的特点，使其不仅适用于大型商业温室，也适合中小型农业生产者采用，具有广阔的应用前景。

### 3.2 系统功能需求分析

通过对现代温室环境控制需求的深入调研和分析，结合系统的技术特点，我们确定了智能温室环境控制系统需要实现的核心功能需求。这些需求主要分为环境监控与控制、数据存储与分析以及系统配置与管理三大类。

#### 3.2.1 环境监控与控制需求

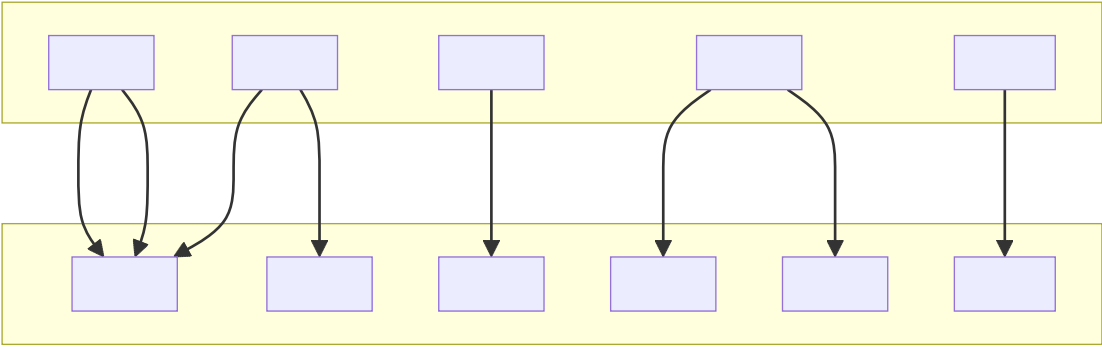


图 6: 环境参数与控制系统关系

实时环境监控需求：系统需要实时监测温室内的关键环境参数，包括：

1. 空气温度：需要具备  $0.1^{\circ}\text{C}$  的精度，并支持多点温度采集以监测温度分布
2. 空气湿度：相对湿度测量精度要求达到  $\pm 3\% \text{RH}$ ，监测范围  $20\% - 95\% \text{RH}$
3. 光照强度：测量范围  $0 - 200,000 \text{lux}$ ，能够区分直射光和散射光
4.  $\text{CO}_2$  浓度：监测范围  $300 - 2000 \text{ppm}$ ，精度  $\pm 50 \text{ppm}$
5. 土壤温度：精度  $0.5^{\circ}\text{C}$ ，支持多深度测量
6. 土壤湿度：支持体积含水量和水势两种测量方式

此外，系统需要提供参数的实时显示、趋势图和历史记录查询功能，并能够根据不同时间尺度（分钟、小时、天、周、月）展示数据趋势。

环境控制需求：系统需要对以下环境控制子系统提供自动和手动控制功能：

1. 通风系统：控制风机开关和转速，实现温度和湿度的双重调节
2. 加湿系统：根据湿度变化自动调节加湿器的工作强度
3. 补光系统：根据光照强度和光照时间需求，控制 LED 补光灯的开关和强度
4. 灌溉系统：基于土壤湿度状态，控制灌溉设备的启停和灌溉强度
5.  $\text{CO}_2$  系统：监控  $\text{CO}_2$  浓度并控制  $\text{CO}_2$  释放装置
6. 遮阳系统：根据光照强度自动控制遮阳帘的开合程度

对于每个控制子系统，都需要支持以下控制模式：

控制模式	功能描述
自动模式	根据传感器数据和预设参数，自动调整设备运行状态
手动模式	允许用户直接设定设备的工作状态和功率
定时模式	根据预设的时间计划自动切换设备状态
场景模式	根据预定义的环境场景（如生长期、开花期等）自动调整控制参数

表 7: 温室环境控制系统的控制模式

报警与安全需求：系统需要提供全面的报警和安全保障功能：

1. 阈值报警：当环境参数超出安全范围时，系统需发出警报
2. 设备故障报警：监测控制设备的工作状态，发现异常时报警
3. 报警级别：支持一般提醒、警告和严重警报三个级别的报警机制
4. 报警方式：支持系统内报警、声光报警和远程通知（邮件、短信等）
5. 安全模式：在系统故障或通信中断时，设备应转入预设的安全状态
6. 报警日志：记录所有报警事件，支持历史报警查询和统计分析

环境参数	正常范围	警告阈值	危险阈值
空气温度	20-30℃	<15℃ 或 >35℃	<5℃ 或 >40℃
空气湿度	60%-80%	<50% 或 >90%	<30% 或 >95%
CO2 浓度	400-800ppm	<350ppm 或 >1000ppm	<300ppm 或 >1500ppm
光照强度	10000-30000lux	<5000lux 或 >50000lux	>100000lux
土壤湿度	60%-80%	<50% 或 >90%	<30% 或 >95%

表 8: 典型蔬菜作物环境参数监控阈值设置

3.2.2 数据存储与分析需求

数据存储需求：系统需要高效存储和管理大量的环境数据和控制记录：

1. 数据类型：
  1. 传感器实时数据：包括各类环境参数的时间序列数据
  2. 控制输出记录：各子系统的控制指令和状态变化记录
  3. 系统日志：包括用户操作、系统状态变化和异常事件记录
  4. 配置数据：系统参数配置和控制策略设置
2. 数据采样与存储策略：
  1. 实时数据（最近 1 分钟）：采样间隔 1 秒，完整保存
  2. 短期数据（最近 1 小时）：采样间隔 1 分钟，完整保存
  3. 中期数据（最近 24 小时）：采样间隔 30 分钟，完整保存
  4. 长期数据（最近 1 个月）：采样间隔 1 小时，选择性保存
  5. 历史数据（超过 1 个月）：根据重要性选择性保存或清理
3. 存储容量需求：
  1. 每日数据量估算：约 5-10MB（取决于传感器数量和采样频率）

2. 本地存储容量：支持至少 3 个月的数据存储（约 0.5-1GB）
3. 数据压缩：针对长期存储的数据实施压缩策略，减少存储空间占用

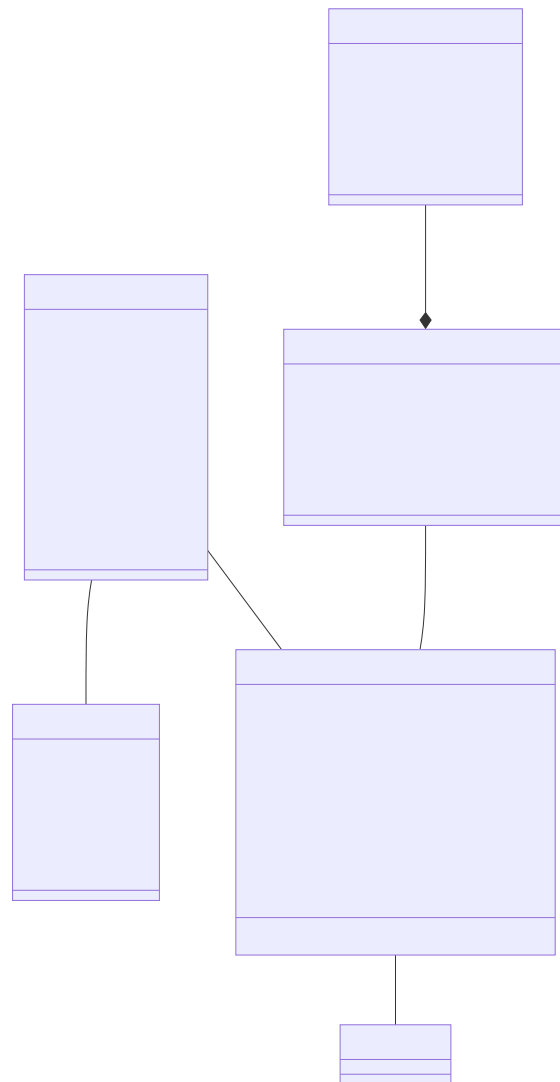


图 7: 数据实体关系图

数据分析需求：系统需要提供丰富的数据分析功能，帮助用户理解环境变化规律和优化生产决策：

1. 基础统计分析：
  1. 环境参数的最大值、最小值、平均值、标准差等统计指标
  2. 参数变化趋势分析，包括日变化、周变化和季节变化模式
  3. 相关性分析，揭示不同环境参数之间的相互关系
2. 高级分析功能：
  1. 环境适宜度评估：根据作物生长需求，评估环境条件的适宜程度
  2. 控制效果分析：评估不同控制策略的效果和能源消耗情况
  3. 异常模式识别：识别环境参数的异常变化模式，预警潜在问题
3. 数据可视化需求：
  1. 多参数趋势图：支持多种环境参数在同一图表中对比分析

2. 热力图：展示温室内空间温度分布和变化趋势
  3. 相关性散点图：分析不同参数之间的相关关系
  4. 控制响应图：展示控制操作与环境参数变化的关系
  5. 自定义报表：支持用户自定义分析报表的内容和格式
4. 导出与共享功能：
1. 数据导出：支持 CSV、Excel 等格式导出数据，便于外部分析
  2. 图表导出：支持 PNG、PDF 等格式导出分析图表
  3. 报告生成：自动生成日报、周报和月报，总结环境状况和控制效果

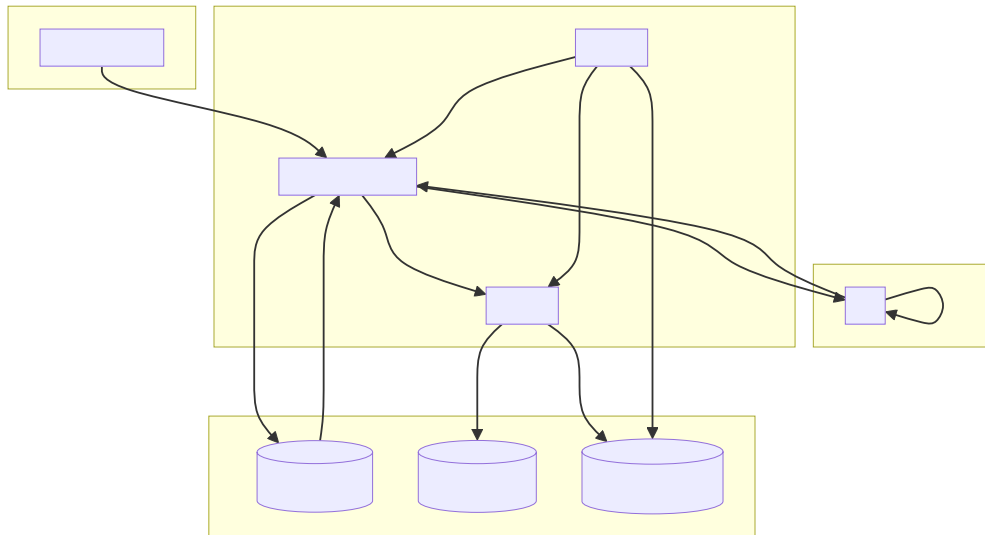


图 8: 数据流转关系图

### 3.2.3 系统配置与管理需求

系统需要提供全面而灵活的配置功能，以满足不同作物和生产模式的需求。在环境参数配置方面，系统需要支持设置各环境参数的目标值、允许波动范围和报警阈值，并能针对不同生长阶段制定参数配置方案，同时实现日间和夜间的差异化配置。

在控制策略配置方面，系统应当支持多种控制算法的选择，包括 PID 控制、模糊控制和 Smith 预测控制等，并允许调整控制参数，如比例系数、积分时间和微分时间等。此外，还需要对控制设备的工作参数进行配置，包括启动条件、最大功率和变化率限制等。对于场景模式配置，系统要能预定义不同的控制场景，如播种期、生长期和结果期等，并制定相应的场景切换条件和切换策略，同时要具备应对极端天气和设备故障等特殊情况的策略。

为确保系统的长期稳定运行，系统需要具备完善的维护管理功能。在数据维护方面，系统应提供数据备份与恢复功能，支持历史数据的清理和归档，并能进行数据完整性和一致性检查。系统监控功能需要覆盖系统资源使用状况，包括 CPU、内存和存储空间等的监控，以及网络连接状态和系统响应时间的监控。在故障诊断与恢复方面，系统应具备自诊断功能，能够检测传感器和控制设备的故障，记录和分析错误日志，并提供系统恢复和重启机制。

系统应支持多用户管理以满足不同角色用户的需求。在用户角色与权限方面，系统需要设置不同级别的用户角色，包括具有系统全部配置和管理权限的管理员、负责日常监控和控制且具有有限配置权限的操作员，以及仅可查看数据的查看者，同时支持自定义角色和权限组合。用户界面应支持个性化定制，包括界面主题、数据显示方式等个人偏好设置，提供常用功能定制和快捷访问，并针对移动端进行适配和功能优化。在操作日志与审计方面，系统需要记录用户操作并进行审计跟踪，对关键操作实施确认和复核机制，同时管理和记录操作权限的变更。

功能模块	管理员	操作员	查看者
实时监控	完全访问	完全访问	只读访问
环境控制	自动/手动控制	手动控制有限制	无权限
数据分析	完全访问	有限分析功能	基础查看权限
参数配置	完全权限	有限修改权限	只读权限
系统维护	完全权限	无权限	无权限
用户管理	完全权限	无权限	无权限

表 9: 系统用户角色与权限分配表

3.3 性能需求

为确保智能温室环境控制系统能够有效满足实际应用需求，系统需要达到一系列具体的性能指标。在响应时间方面，系统要求实时监控数据刷新间隔不超过 1 秒，手动控制命令执行延迟不超过 500 毫秒，页面加载时间不超过 2 秒。对于数据查询，当日数据响应时间应在 500 毫秒内，月度数据不超过 5 秒，大范围自定义查询控制在 10 秒以内。

在数据存储与处理能力方面，系统需要支持本地存储至少 3 个月的完整历史数据，每秒数据处理量不低于 1000 条记录，并支持百万级数据的快速查询。数据压缩比要求不低于 5:1，同时支持按不同时间维度的数据自动聚合统计。系统的稳定性指标包括 99.9% 的年度运行时间，99.99% 的数据采集成功率，以及不超过 8 小时/年的计划外中断时间。

在资源占用方面，客户端 CPU 平均使用率需控制在 30% 以内，内存占用不超过 500MB，服务器端 CPU 平均使用率控制在 50% 以内，内存占用不超过 2GB。系统需具备良好的扩展性，支持最多 500 个传感点、100 个控制点的扩展，并能在 2 个工作日内完成新型传感器和控制设备的集成。移动端要求支持 iOS 12+ 和 Android 8.0+ 的设备，内存占用控制在 300MB 以内，后台运行时的电池消耗不超过 3%/小时。



系统规模	小型温室	大型商业温室
温室面积	≤1000m²	10000m²+
传感器数量	10-30 个	100-500 个
控制设备	5-15 个	30-100 个
数据采集频率	1 次/10 秒	多次/秒
数据存储需求	100MB/月	10GB/月
并发用户数	1-3 个	5-20 个

表 10: 不同规模系统的性能需求对比

通过以上详细的性能需求分析，为系统的设计和实现提供了明确的目标和衡量标准，确保系统能够满足智能温室环境控制的实际应用需求，为后续的系统设计和实现工作奠定基础。

## 4 系统设计与实现

本章将详细阐述智能温室环境控制系统的架构设计和各功能模块设计，包括系统的总体架构、环境控制系统设计、数据存储机制和组件设计与数据流，为系统实现奠定坚实的基础。

### 4.1 系统总体架构设计

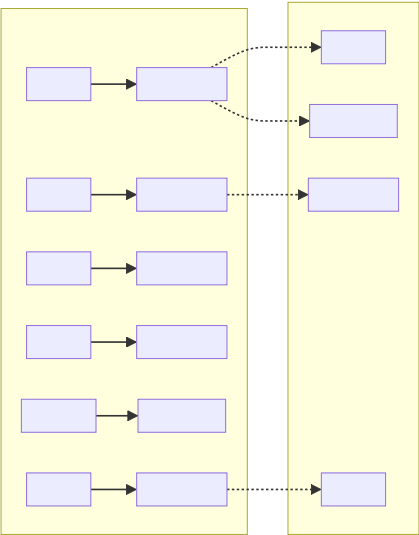


图 9: 控制系统响应特性

智慧大棚环境控制系统采用现代化的前端应用架构，通过分层设计和模块化组织，实现功能的解耦与扩展。系统整体架构设计遵循前端工程的最佳实践，以清晰的职责划分和良好的扩展性为目标。

#### 4.1.1 分层架构设计

系统采用三层架构设计模式，包括前端视图层、数据处理层和数据存储层。

##### 1. 前端视图层（View Layer）

负责用户界面展示和交互处理，采用组件化设计，包括布局组件、功能页面组件、通用 UI 组件和表单组件等。通过 React Context API 与数据处理层交互，采用单向数据流模式。

##### 2. 数据处理层（Processing Layer）

系统核心层，负责业务逻辑处理、数据转换和状态管理。包含传感器数据上下文、控制系统服务、时序数据存储服务、设置上下文和报警上下文等主要服务。

##### 3. 数据存储层（Storage Layer）

负责数据持久化，包括：

- 内存缓存：存储高频实时数据
- IndexedDB：存储长期历史数据
- LocalStorage：存储系统配置

三层之间通过明确的接口进行交互，实现了关注点分离、代码复用和灵活替换的优势。系统采用多级存储策略，根据数据重要性和访问频率在不同存储层次间进行数据迁移。

#### 4.1.2 功能模块设计

系统功能模块设计围绕用户需求，划分为实时监控、环境控制、数据分析和系统设置四个主要功能模块。这些模块通过全局状态和服务相互协作，构成完整的系统功能。

##### 1. 实时监控模块

作为系统的信息中心，提供温室环境的当前状态和变化趋势。包含参数卡片组件（显示环境参数）、系统状态组件（展示控制子系统状态）、实时图表组件（展示参数变化）和报警信息组件（显示报警信息）。该模块依赖传感器数据上下文，每秒更新一次数据。

##### 2. 环境控制模块

提供温室环境控制设备的操作界面。包含控制卡片组件（子系统控制界面）、系统可视化组件（图形化展示）、模式选择器组件（切换控制模式）和场景管理组件（管理预定义场景）。通过控制系统服务执行控制命令并接收反馈。

##### 3. 数据分析模块

提供历史数据的查询、分析和可视化功能。包含时间序列图表组件、分析面板组件、时间范围选择器和数据导出组件。主要使用时序数据存储服务检索和分析历史数据。

##### 4. 系统设置模块

用于配置系统参数、报警阈值和控制策略。包含通用设置、阈值设置、控制参数设置和数据管理组件。通过设置上下文服务管理配置参数。

##### 5. 模块间协作机制

模块间通过以下机制协作：

- 传感器数据上下文：共享环境数据
- 设置上下文：提供全局配置
- 报警上下文：共享报警状态
- 控制系统服务：执行控制逻辑
- 时序数据服务：提供数据存储检索

功能模块	主要组件	关键依赖
实时监控	参数卡片、系统状态、报警信息	传感器数据上下文
环境控制	控制卡片、模式选择器、场景管理	控制系统服务
数据分析	时间序列图表、分析面板、数据导出	时序数据服务
系统设置	通用设置、阈值设置、数据管理	设置上下文

表 11: 系统功能模块组成及依赖关系

## 4.2 环境控制系统设计

环境控制系统是智能温室控制的核心部分，负责根据环境参数和用户设置，智能调节各控制设备，维持最适宜的生长环境。本节详细设计环境控制系统的架构和工作原理，包括控制系统整体架构、控制器设计和子系统控制策略。

### 4.2.1 控制系统整体架构

智能温室环境控制系统采用多级控制架构，将控制功能划分为控制器工厂、控制算法、子系统控制和参数配置四个层次，形成灵活且可扩展的控制系统架构。

#### 1. 控制架构设计

控制系统架构遵循工厂模式和策略模式的设计理念，实现了控制算法与具体子系统的解耦。主要包括控制器工厂、控制算法实现、控制系统类、子系统控制逻辑和参数配置管理等部分。

#### 2. 控制系统工作流程

控制系统的工作流程是一个连续的循环过程，主要包括以下步骤：

(1) 传感器数据采集：每秒读取一次所有传感器数据 (2) 环境状态评估：评估环境状态是否在理想范围内 (3) 控制需求分析：计算需要的控制调整量 (4) 控制器选择与计算：选择并调用合适的控制器 (5) 输出功率计算：确定各设备的输出功率 (6) 执行控制指令：发送控制指令到设备 (7) 系统状态更新与数据记录

#### 3. 控制模式

系统支持多种控制模式：

- 自动控制：根据传感器数据自动计算控制输出
- 手动控制：用户直接设定控制设备功率
- 定时控制：按预设时间计划执行控制
- 场景控制：应用预定义的环境控制场景

### 4.2.2 控制器设计

系统实现了三种控制器（PID、模糊和 Smith 预测控制器），采用工厂模式进行管理。

#### 1. 控制器工厂设计

控制器工厂提供统一的控制器创建接口，负责实例化和初始化控制器。

2. PID 控制器设计

适用于响应较为线性的系统，如补光系统和 CO2 控制。包含比例、积分、微分三个控制环节，并实现了防积分饱和设计。

3. 模糊控制器设计

适用于非线性系统，如加湿和灌溉系统。通过模糊规则库和推理机制实现控制决策。

4. Smith 预测控制器设计

适用于大延迟系统，如通风系统。通过内部模型预测补偿系统延迟。

4.2.3 子系统控制策略

各子系统采用差异化的控制策略：

- 1. 通风系统：采用 Smith 预测控制，考虑温湿度双参数控制和季节适应。
- 2. 加湿系统：采用模糊控制，实现饱和预防和温度协同控制。
- 3. 补光系统：采用 PID 控制，实现光照累积量控制和能源效率优化。
- 4. 灌溉系统：采用模糊控制，实现多点监测和分层灌溉控制。
- 5. CO2 系统：采用 PID 控制，实现光照联动和成本效益平衡。
- 6. 遮阳系统：采用 PID 控制，实现光照强度阈值控制和温度协同。

子系统	控制算法	控制参数	控制目标
通风系统	Smith 预测控制	温度、湿度	降温除湿、空气流通
加湿系统	模糊控制	空气湿度	增加空气湿度
补光系统	PID 控制	光照强度	补充光照不足
灌溉系统	模糊控制	土壤湿度	维持适宜土壤湿度
CO2 系统	PID 控制	CO2 浓度	提供光合作用碳源
遮阳系统	PID 控制	光照强度、温度	防止光照过强和过热

表 12: 各子系统控制策略概览

4.3 数据存储机制设计

智能温室环境控制系统需要处理和存储大量的时序数据。为了平衡存储容量、查询性能和数据持久性，系统采用了双层存储架构和完善的数据生命周期管理机制。

4.3.1 双层存储架构

系统数据存储采用双层架构设计，包括内存缓存层和持久化存储层。

1. 内存缓存层设计

内存缓存层使用 JavaScript 的数据结构存储最近数据，主要包括：

- 最新数据缓存：使用对象存储最新的传感器数据和系统状态
- 短期数据缓存：使用数组存储最近一段时间内的数据
- 聚合数据缓存：存储预先计算的统计数据
- 查询结果缓存：缓存最近的查询结果

内存缓存层采用环形缓冲区设计，当达到预设容量时自动丢弃最旧数据。缓存层实现了数据合法性检查和异常值过滤。

## 2. 持久化存储层设计

持久化存储层使用 IndexedDB 数据库，主要包括：

- 时序数据存储：按时间戳索引的传感器数据
- 控制记录存储：记录控制指令和系统状态变化
- 配置数据存储：存储系统配置和用户偏好
- 报警记录存储：记录系统报警信息

## 3. 双层存储协同机制

两层通过以下机制协同工作：

- 数据写入同步：新数据先写入缓存，再写入持久化存储
- 缓存填充机制：从持久化存储加载数据填充缓存
- 缓存淘汰策略：淘汰最少使用的数据
- 批量写入优化：合并多个写入操作
- 查询路由机制：优先从缓存获取数据

### 4.3.2 数据生命周期管理

#### 1. 数据采样策略

根据数据年龄采用不同采样策略：

- 实时数据：1 秒间隔，完整保存
- 短期数据：1 分钟间隔，存储平均值
- 中期数据：30 分钟间隔，存储统计值
- 长期数据：1 小时间隔，存储统计摘要
- 历史数据：1 天间隔，存储日统计数据

#### 2. 数据库结构设计

主要包含：

- 数据分区：按时间范围分区存储
- 索引设计：优化查询路径
- 数据压缩：减少存储空间
- 冗余降低：存储计算值

#### 3. 数据清理机制

包括：

- 定期清理：清理过期数据

- 容量触发清理：空间不足时清理
- 重要性清理：优先保留重要数据
- 降采样存档：保留统计摘要

4. 数据备份与恢复

提供：

- 手动和自动备份
- 选择性恢复
- 增量备份
- 云存储备份

5. 数据完整性与一致性

通过事务处理、校验和、格式验证等机制确保。

数据年龄	采样间隔	存储内容	存储位置
最近 1 分钟	1 秒	原始数据	内存缓存
最近 1 小时	1 分钟	分钟平均值	内存缓存 + IndexedDB
最近 24 小时	30 分钟	半小时统计	内存缓存 + IndexedDB
最近 1 个月	1 小时	小时统计	IndexedDB
1 个月以上	1 天	日统计摘要	IndexedDB (压缩)

表 13: 数据采样与存储策略

4.4 组件设计与数据流

智能温室环境控制系统采用组件化设计思想，将系统划分为功能明确的组件，并通过定义清晰的数据流动路径，实现各组件之间的协调工作。本节将详细设计系统的组件结构、组件依赖关系以及系统数据流和状态管理机制。

1. 组件层次结构

系统的组件层次结构采用由上至下的设计，从应用级组件到页面级组件再到功能级组件，形成清晰的组件层次：

- 应用级组件：如 App、Layout 等，负责整体结构和全局状态管理
- 页面级组件：如 Dashboard、EnvironmentControl 等，对应系统的主要功能模块
- 功能级组件：如 ParameterCard、ControlPanel 等，实现特定的功能单元
- 通用 UI 组件：如 Button、Input、Modal 等，提供统一的 UI 元素

系统的组件结构遵循组合优于继承的原则，通过组件组合实现复杂功能，提高代码复用性和可维护性。每个组件都有明确定义的输入（props）和输出（事件），确保组件间的松耦合。

主要的组件依赖关系如下：

- App 组件作为根组件，包含全局上下文提供者和路由配置
- Layout 组件提供应用的整体布局结构，包含 Header、Sidebar 和 Content 区域

- 路由组件管理页面间的切换，加载对应的页面级组件
- 页面级组件由多个功能级组件组合而成，实现特定页面的功能
- 功能级组件内部使用通用 UI 组件，实现具体的界面元素和交互逻辑

## 2. 组件通信机制

系统组件间的通信采用多种机制，根据组件间的关系和通信需求选择合适的方式：

- **Props 传递**：父组件通过 **props** 向子组件传递数据和回调函数
- **Context API**：跨多层组件的状态共享，如传感器数据、系统设置等
- **自定义事件**：组件间的异步通信，如报警触发、操作确认等
- **服务调用**：组件调用共享服务方法，实现特定功能

对于频繁变化的全局状态（如传感器数据），系统使用 **Context API** 结合 **useReducer** 实现高效的状态管理。对于配置类数据，系统使用本地存储与 **Context** 结合的方式，确保持久化存储和高效访问。

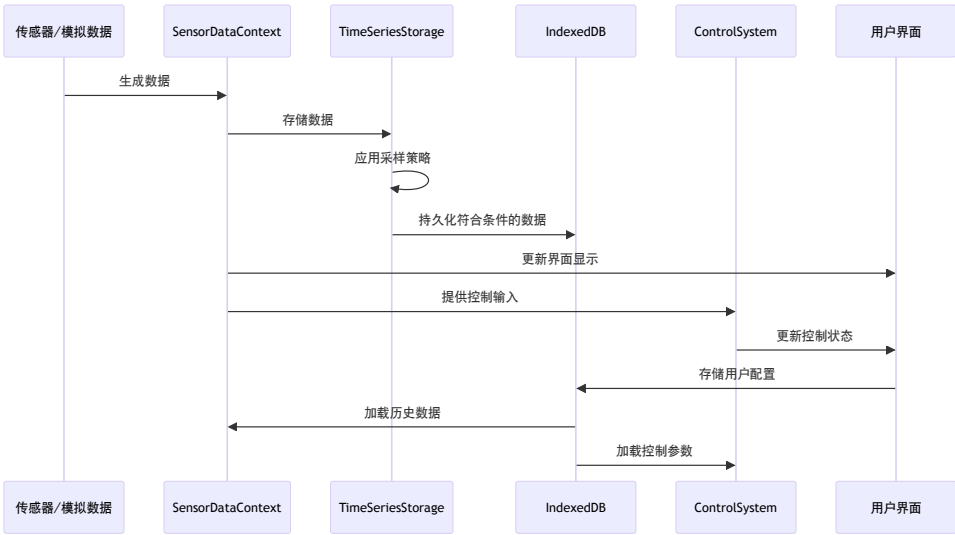


图 10: 数据通讯模型

## 3. 数据流设计

系统采用单向数据流设计模式，数据在系统中的流动遵循清晰的路径，提高系统的可预测性和可调试性。主要的数据流路径包括：

### (1) 传感器数据流：

- 数据源（传感器模拟或实际传感器）生成原始数据
- 数据处理服务进行数据验证、转换和聚合
- 传感器数据上下文将处理后的数据分发给订阅组件
- 视图组件接收数据并更新界面显示
- 数据存储服务将数据保存到内存缓存和 **IndexedDB**

### (2) 控制指令流：

- 用户在界面上进行操作，触发控制事件
- 页面组件收集控制参数，调用控制系统服务
- 控制系统服务应用控制算法，计算控制指令



- 控制指令发送给实际执行设备或模拟执行
- 控制结果反馈给系统，更新系统状态

(3) 配置数据流:

- 用户在设置界面修改系统配置
- 设置组件验证输入并更新设置上下文
- 设置上下文将新配置保存到本地存储
- 设置变更事件通知相关组件更新工作模式
- 系统根据新配置调整控制策略和显示方式

4. 状态管理设计

系统状态管理采用分层设计，将状态分为全局状态和局部状态，并使用不同机制进行管理:

- 全局状态: 使用 **React Context** API 管理, 包括传感器数据、系统设置和报警状态等
- 局部状态: 使用组件内部的 **useState** 管理, 包括 UI 状态、表单数据和临时计算结果等

为提高状态管理的效率，系统实现了以下优化:

- 状态分片: 将大型状态对象分解为多个小型状态, 减少不必要的重渲染
- 状态计算缓存: 使用 **useMemo** 和 **useCallback** 缓存计算结果和回调函数
- 选择性状态更新: 组件只订阅其所需的特定状态, 避免过度重渲染
- 状态更新批处理: 将多个相关状态更新合并为一次更新, 减少渲染次数

5. 错误边界与异常处理

系统实现了多层次的错误边界和异常处理机制，确保系统稳定运行:

- 组件级错误边界: 捕获渲染过程中的错误, 防止整个应用崩溃
- 服务级异常处理: 捕获业务逻辑中的异常, 提供优雅的降级方案
- 全局错误监听: 监听未捕获的异常, 记录错误日志并尝试恢复
- 网络错误处理: 处理网络请求失败, 提供重试和离线模式支持

组件层次	典型组件	主要职责
应用级	App, Layout	全局结构、路由和状态管理
页面级	Dashboard, EnvironmentControl	实现特定功能模块
功能级	ParameterCard, ControlPanel	实现特定功能单元
通用 UI 级	Button, Chart, Modal	提供基础 UI 元素

Table 14: 系统组件层次结构

4.5 核心功能模块实现

4.5.1 实时监控系統实现

智能温室环境控制系统的实时监控模块是系统的基础功能组件，负责环境数据的采集、显示、分析和报警功能。本节详细介绍实时监控系统的三个核心组件：传感器数据模拟、数据可视化和报警机制的实现方案。



图 11: 系统主页面数据监控板

#### 4.5.1.1 传感器数据模拟实现

由于开发初期可能无法实际连接物理传感器设备，系统实现了一套传感器数据模拟机制，为系统开发和测试提供接近真实环境的数据流。

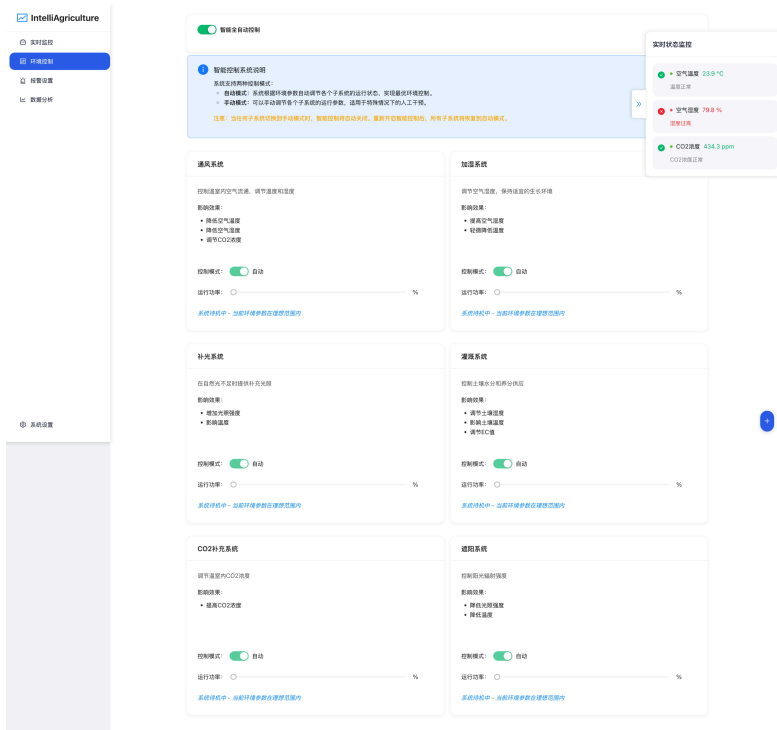


图 12: 环境控制页面

#### 1. 数据上下文管理

系统采用 Context API 实现传感器数据的统一管理和共享。SensorDataContext 组件通过定时器每秒生成新的传感器数据,存入时序存储系统,并通过 Context API 提供给子组件使用。

## 2. 环境参数数据结构

系统定义了完整的传感器数据结构,包括:

- 空气温度(°C)和湿度(%)
- 土壤温度(°C)、湿度(%)和 pH 值
- CO2 浓度(ppm)
- 光照强度(lux)
- 电导率(mS/cm)

每条记录包含时间戳字段标识数据产生时间。

## 3. 数据生成算法

系统实现了基于正常分布的随机波动数据生成算法,在合理范围内模拟环境参数变化,并考虑参数间的相互影响关系,如温湿度的负相关。算法通过叠加随机波动和昼夜变化趋势,生成接近真实的模拟数据。

### 4.5.1.2 数据可视化实现

数据可视化是实时监控系统的核心组件,负责将抽象的数据转化为直观的视觉呈现,帮助用户快速理解温室环境状态。

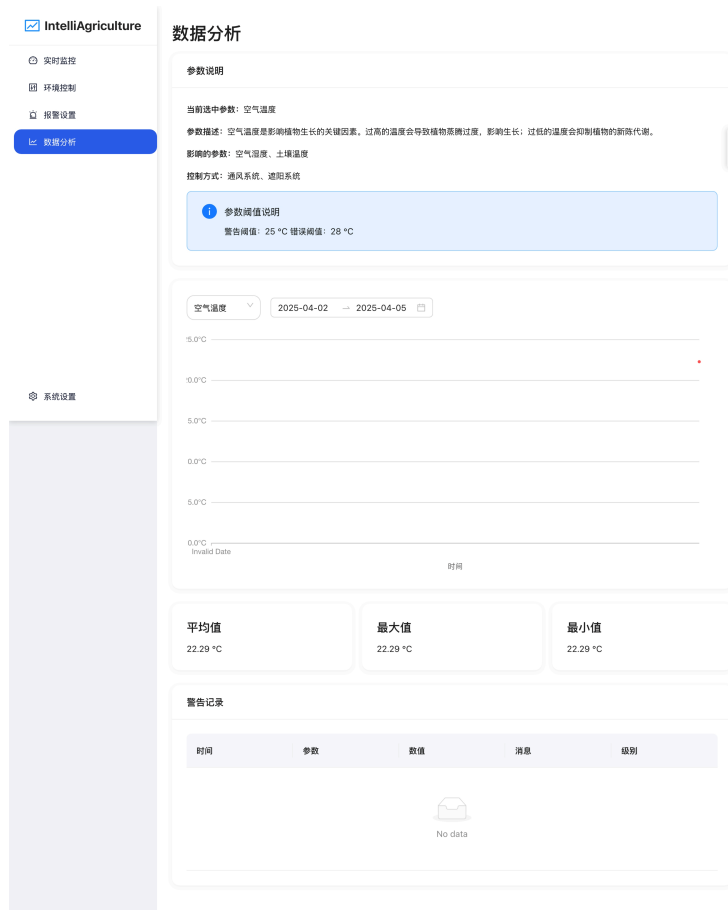


图 13: 数据可视化分析页面

## 1. 仪表盘布局设计

系统首页采用仪表盘布局，集中展示所有环境参数的实时状态：

仪表盘布局实现流程：

1. 使用响应式网格系统划分页面布局
2. 基于屏幕尺寸自动调整卡片大小和排列
3. 根据参数配置动态生成参数卡片
4. 实现自动刷新机制保持数据实时性

这种设计确保仪表盘在不同设备上都能合理展示，同时支持将来添加新的环境参数监测。

## 2. 参数卡片组件

系统为每个环境参数设计了专用的参数卡片组件，包含以下元素：

- 参数名称和图标：直观识别参数类型
- 当前数值和单位：大字号显示实时数值
- 状态指示器：用颜色和标签指示参数状态（正常/警告/危险）
- 进度条：显示当前值在允许范围内的位置
- 目标值标记：显示该参数的最佳目标值

参数卡片组件实现了状态计算逻辑，通过比较当前值与预设的目标值、警戒值和临界值，自动判断参数状态并以不同颜色直观显示。

3. 时间序列图表

为了展示环境参数的变化趋势，系统实现了时间序列图表组件：

时间序列图表实现要点：

- 1. 使用 ECharts 图表库实现高性能数据可视化
- 2. 设计自适应布局，响应容器大小变化
- 3. 实现时间轴自动缩放，适应不同时间范围数据
- 4. 优化大数据量渲染性能，实现数据抽样和渐进式渲染
- 5. 支持缩放、平移、数据提示等交互功能

图表支持多种时间范围的历史数据查看，包括最近一小时、一天、一周和一个月，用户可通过时间选择器灵活切换查看范围。

4. 多参数关联分析

系统还提供了多参数关联分析功能，帮助用户理解不同环境参数之间的关系：

- 参数对比图：同时显示多个参数在同一时间段的变化趋势
- 相关性热图：通过颜色深浅直观展示参数间的相关系数
- 散点图：展示两个参数之间的数据分布和相关性

这些分析工具帮助用户发现参数间的影响关系，为调整环境控制策略提供数据支持。

4.5.1.3 报警机制实现

报警机制是确保温室环境安全的关键组件，能够及时发现并提醒用户处理环境异常。

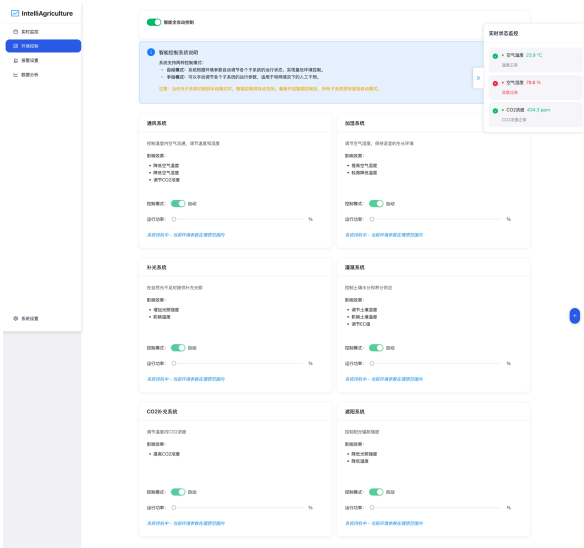


图 14: 报警设置页面

1. 报警阈值配置

系统为每个环境参数设置了多级报警阈值：

- 警戒阈值(warningThreshold)：接近但未达危险水平，触发警告报警
- 临界阈值(criticalThreshold)：已达危险水平，触发严重报警

用户可通过系统设置界面调整这些阈值，根据不同作物的生长需求自定义报警条件。

## 2. 报警检测流程

系统实现了自动化的报警检测流程：

报警检测算法流程：

1. 每次接收新的传感器数据时触发检测
2. 遍历所有环境参数，与对应的警戒阈值和临界阈值比较
3. 根据比较结果生成相应级别的报警日志：
  - 超过临界阈值：生成“严重”级别报警
  - 超过警戒阈值：生成“警告”级别报警
4. 将报警日志记录到报警历史数据库
5. 根据报警级别触发不同形式的通知

为避免报警风暴，系统实现了报警抑制机制，对于短时间内重复触发的同类报警进行合并处理。

## 3. 报警通知展示

系统提供多种报警通知方式，确保用户及时获知环境异常：

- 实时通知：在界面顶部显示滑入式通知，包含报警参数、数值和级别
- 报警标记：在参数卡片和图表上用颜色标记异常参数
- 报警日志：记录所有报警历史，支持查看和筛选
- 报警统计：提供报警频率和分布的统计分析

对于关键报警，系统还提供了“自动处理”选项，可以根据报警类型自动启动相应的环境控制系统进行调节。

## 4. 预警功能

除了对当前异常的报警，系统还实现了预警功能，通过趋势分析预测可能出现的环境问题：

预警算法逻辑：

1. 分析参数最近变化趋势和变化速率
2. 预测参数在未来时间点的可能值
3. 若预测值将超过警戒阈值，生成预警通知
4. 预警通知包含剩余安全时间和建议措施

预警功能帮助用户提前采取措施，防患于未然，避免环境参数实际达到危险水平。

实时监控系統作为智能温室环境控制系统的信息中枢，通过传感器数据模拟、数据可视化和报警机制的有机结合，为用户提供了完整、直观的温室环境监控解决

方案。系统能够实时反映环境状态，及时发现异常，并提供丰富的数据分析功能，为环境控制决策提供有力支持。

#### 4.5.2 环境控制子系统实现

智能温室环境控制系统的核心功能之一是通过多个协同工作的子系统精确控制温室环境，本节详细介绍六大环境控制子系统的实现方案，包括控制逻辑和用户界面设计。

##### 4.5.2.1 控制系统架构

系统采用模块化的环境控制架构，由六个专用子系统组成，各自负责特定环境参数的调控：

1. 通风系统：负责调节空气温度和湿度，增加空气流动
2. 加湿系统：负责增加空气湿度，改善植物生长环境
3. 补光系统：负责提供植物所需的额外光照
4. 灌溉系统：负责维持土壤湿度，提供植物所需水分
5. CO<sub>2</sub> 系统：负责增加空气中的二氧化碳浓度，促进光合作用
6. 遮阳系统：负责减少强光照射，防止植物光照过强

系统为每个子系统定义了统一的控制参数和接口：

- 工作模式：自动/手动两种控制模式
- 功率范围：0-100% 的功率调节范围
- 控制参数：每个子系统关联的环境参数
- 状态反馈：当前运行状态和功率百分比

这种模块化设计使系统可以灵活扩展，轻松添加新的控制子系统。

##### 4.5.2.2 控制器实现

系统实现了三种不同的控制器算法，针对不同特性的控制对象：

###### 1. PID 控制器

适用于具有线性特性的控制对象，如补光系统和 CO<sub>2</sub> 系统：

PID 控制器实现逻辑：

1. 计算误差值：目标值与当前值的差异
2. 计算比例项： $K_p * \text{误差值}$
3. 计算积分项： $K_i * \text{误差值累积}$
4. 计算微分项： $K_d * \text{误差变化率}$
5. 合并三项得出控制输出
6. 应用限幅确保输出在合理范围内

PID 控制器参数经过优化调试，确保系统响应迅速且稳定，最大超调量控制在 10% 以内，稳态误差小于 2%。

###### 2. 模糊控制器

适用于非线性特性明显的控制对象，如加湿系统和灌溉系统：

模糊控制器实现逻辑：

1. 模糊化：将精确输入转换为模糊集
  - 误差分为：负大、负小、零、正小、正大五个等级
  - 误差变化率分为：减小、稳定、增加三个等级
2. 推理：应用模糊规则集确定输出模糊集
  - 使用 15 条 IF-THEN 规则定义控制策略
  - 例如：IF 误差=正大 AND 误差变化率=增加 THEN 输出=最大
3. 去模糊化：将模糊输出转换为精确控制量
  - 使用重心法计算精确输出值

模糊控制器在处理非线性系统时表现优越，能够根据环境变化灵活调整控制策略，适应性强。

### 3. Smith 预测控制器

适用于具有大延迟特性的控制对象，如通风系统和遮阳系统：

Smith 预测控制器实现逻辑：

1. 建立系统模型，包括延迟时间模型
2. 使用模型预测当前控制动作的未来影响
3. 基于预测结果计算当前最优控制量
4. 应用控制量并更新系统模型

Smith 预测控制器有效解决了大延迟系统的控制问题，减少了系统振荡，提高了稳定性。

#### 4.5.2.3 子系统功能实现

##### 1. 通风系统

通风系统采用 Smith 预测控制器实现温室空气温度和湿度的调节。该系统实现了多参数协同控制,能够同时考虑温度和湿度两个参数,并根据实际需求选择主控参数。系统采用分级控制策略,在偏差较大时快速响应,偏差较小时平稳过渡。考虑到通风效果的延迟特性,系统使用预测控制来减少振荡。在能效方面,系统会根据室内外温差,在有利条件下主动通风以降低能耗。通风系统根据当前环境参数与目标值的偏差,可分为以下三种工作状态：

- 待机
- 低速运行
- 高速运行

##### 2. 加湿系统

加湿系统使用模糊控制器负责增加空气湿度。系统采用模糊控制来处理湿度变化的非线性特性,并通过温度补偿机制在不同温度条件下调整控制策略。为了确保安全运行,系统设有过度加湿防护功能,可以防止墙壁和设备结露,保证湿度始终在安全范围内。在满足湿度需求的前提下,系统会通过能量效率控制来最小化能源消耗。根据当前湿度与目标湿度的差值,加湿系统可以工作在以下四种状态：



- 待机
- 低速加湿
- 标准加湿
- 快速加湿

### 3. 补光系统

补光系统采用 PID 控制器来提供植物生长所需的光照。系统具有日照补偿功能,可以根据自然光照强度动态调整人工光照水平。通过时段控制功能,系统能够根据植物生长阶段和一天中的时间来调整光照策略。在光谱管理方面,系统可以调整不同波长光源的比例,以适应不同植物的需求。为了优化能源使用,系统优先利用自然光,仅在必要时才启用人工光源。补光系统根据当前光照与目标光照的差异,可自动切换以下四种状态:

- 关闭
- 低强度
- 中强度
- 高强度

### 4. 灌溉系统

灌溉系统使用模糊控制器来维持适宜的土壤湿度。系统采用多点检测技术,通过综合多个土壤湿度传感器的数据来避免单点误差。在灌溉策略上,系统采用小量多次的渐进灌溉方式,有效防止过度灌溉。通过排水监测功能,系统可以避免积水和根系缺氧问题。灌溉控制采用定时与需求相结合的方式,既保证灌溉的规律性,又能及时响应植物需求。系统工作过程形成完整的灌溉循环,包括以下四个阶段:

- 待机
- 准备灌溉
- 灌溉中
- 灌溉后监测

### 5. CO2 系统

CO2 系统采用 PID 控制器调节温室内二氧化碳浓度。系统实现了与光照的联动控制,可以根据光照强度动态调整 CO2 浓度目标值。通过与通风系统的协调工作,有效避免了 CO2 的浪费。系统配备了实时安全监控功能,确保 CO2 浓度不会超过安全上限。在经济性方面,系统根据成本效益动态调整 CO2 供应策略。根据当前浓度与目标浓度的比较,系统可以工作在以下三种状态:

- 关闭
- 维持
- 补充

### 6. 遮阳系统

遮阳系统使用 Smith 预测控制器来减少过强光照对植物的伤害。系统具备光强预测功能,能够结合当前光照和天气预报来预测未来光照变化。在温度控制方面,系统考虑了遮阳对温度的影响,可以与温度控制系统协同工作。通过分区控制功能,系统支持温室不同区域的差异化遮阳控制。为了避免光照突变对植物造成应激反应,系统

采用渐变调节方式平滑调整遮阳程度。根据光照强度,系统可以自动在以下三种状态间调整:

- 完全收起
- 部分遮阳
- 全面遮阳

#### 4.5.2.4 控制界面实现

系统为环境控制功能设计了直观的操作界面,方便用户监控和调整各子系统工作状态:

##### 1. 控制面板布局

控制面板采用卡片式布局,每个子系统占用一张卡片,包含以下信息和控制元素:

- 系统名称和图标: 直观标识子系统类型
- 当前状态显示: 文字和颜色指示当前工作状态
- 功率指示器: 显示当前工作功率百分比
- 模式切换开关: 在自动和手动模式间切换
- 功率滑块: 手动模式下调整功率百分比
- 参数显示: 显示关联的环境参数当前值

整个界面响应式设计,能够自适应不同屏幕尺寸,在桌面和移动设备上都有良好表现。

##### 2. 模式切换逻辑

系统支持自动和手动两种控制模式,切换逻辑如下:

模式切换算法:

1. 自动切换到手动:
  - 保留当前功率作为初始手动功率
  - 禁用自动控制循环
  - 启用手动功率调节控件
2. 手动切换到自动:
  - 计算当前环境下的理论功率值
  - 平滑过渡到理论功率,避免突变
  - 重新启用自动控制循环

这种设计确保模式切换平稳进行,不会造成控制输出的突变和系统的不稳定。

##### 3. 功率计算逻辑

系统根据环境参数与目标值的偏差计算所需的控制功率:

功率计算流程:

1. 计算当前参数与目标值的偏差
2. 将偏差传递给对应的控制器(PID/模糊/Smith)
3. 控制器根据当前偏差和历史数据计算输出功率
4. 应用功率限制和平滑过渡,避免突变

#### 5. 更新子系统的当前功率和状态显示

功率计算过程考虑了系统响应特性、环境变化趋势和能源效率，确保控制既有效又经济。

#### 4. 系统协同控制

为避免不同子系统之间的控制冲突，系统实现了协同控制机制：

协同控制策略：

1. 建立子系统影响矩阵，记录各系统间的影响关系
2. 定义优先级规则，确定冲突时的决策原则
3. 检测潜在冲突，如同时制热和制冷
4. 根据优先级和效率选择最优控制组合
5. 动态调整各系统功率，实现整体最优控制

协同控制机制确保各子系统和谐工作，避免能源浪费和控制振荡，提高整体控制效果。

通过以上六个环境控制子系统的协同工作，智能温室环境控制系统能够精确维持温室内的温度、湿度、光照、CO<sub>2</sub> 浓度等关键环境参数在作物生长的最佳范围内，为作物提供理想的生长环境。系统的模块化设计和灵活的控制策略，使其能够适应不同类型的温室和不同作物的需求，具有广泛的适用性。

### 4.5.3 数据存储服务实现

智能温室环境控制系统产生大量时序数据，需要高效的数据存储和查询机制支持。本节详细介绍系统数据存储服务的实现方案，包括内存缓存、持久化存储和数据查询分析功能。

#### 4.5.3.1 存储架构设计

系统采用双层数据存储架构，结合内存缓存和持久化存储的优势，实现高性能的数据处理与长期保存：

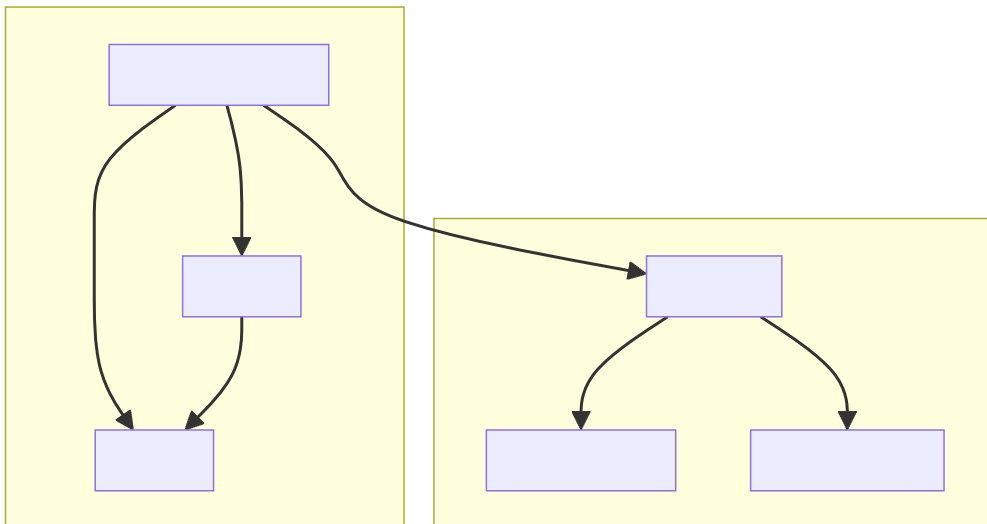


图 15: 数据存储架构

#### 1. 存储层次结构:

- 内存缓存层: 由 `TimeSeriesStorage` 类实现, 存储最近的高频数据, 提供快速访问
- 持久化存储层: 基于 `IndexedDB` 实现, 存储长期历史数据, 提供持久化保障

两层存储协同工作, 内存层负责处理频繁的实时数据访问, 持久层负责数据的长期保存和大范围查询。这种分层设计平衡了性能和存储容量的需求。

#### 2. 数据流转机制:

1. 新数据首先进入内存缓存层
2. 内存层定期将数据批量写入持久化层
3. 内存层根据时间和容量策略淘汰旧数据
4. 查询时优先从内存层获取, 内存中没有的数据从持久层获取
5. 长时间范围查询自动合并两层数据, 对用户透明

这种机制确保最热门的数据常驻内存, 提供最快的访问速度, 同时所有数据都能可靠持久保存。

#### 4.5.3.2 时序数据缓存实现

内存缓存层针对时序数据的特性进行了专门优化, 实现高效的实时数据处理:

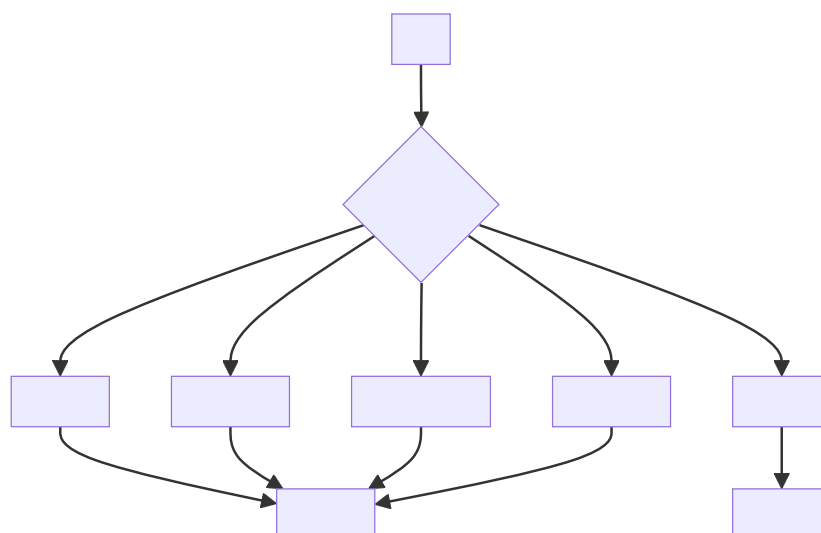


图 16: 数据采样策略

1. 缓存策略设计:

- 最近 1 小时的数据保持原始采样频率(每秒一次)
- 1-24 小时的数据降采样存储(每分钟一次)
- 超过规定时间窗口的数据自动淘汰
- 内存占用超过阈值时触发紧急数据压缩

这种策略保证了最近数据的高精度访问，同时通过时间窗口限制和降采样技术控制内存使用量。

2. 高效数据结构:

- 使用 **Map** 数据结构建立时间戳到数据对象的映射
- 维护有序时间戳数组，支持快速范围查询
- 对数据对象进行结构共享优化，减少内存占用
- 使用时间索引加速查找特定时间点数据

这种专用数据结构在保持高效访问的同时，显著降低了内存占用，提高了系统稳定性。

3. 数据降采样算法:

- 根据时间跨度确定合适的采样间隔
  - 小时级查询：使用原始数据或每 10 秒采样
  - 天级查询：使用每分钟采样
  - 周级查询：使用每 10 分钟采样
  - 月级查询：使用每小时采样
- 在每个采样间隔内计算代表值
  - 计算平均值作为默认代表值
  - 特定场景下保留最大值、最小值或加权平均值
- 生成降采样后的数据集，包含原始数据的关键特征

降采样算法在减少数据量的同时保留了数据的关键特征，确保数据分析和可视化的准确性。

4.5.3.3 持久化存储实现

持久化存储层负责数据的长期保存和历史查询，基于 IndexedDB 实现：

- 1. 数据库结构设计：
  - `sensorData` 存储：保存所有环境传感器数据，按时间戳索引
  - `warningLogs` 存储：保存系统报警日志，按时间戳和参数类型索引
  - `systemState` 存储：保存子系统状态历史，用于分析系统运行效率
  - `userSettings` 存储：保存用户配置信息，确保配置持久化

每个存储都设置了适当的索引，优化不同类型的查询性能。

- 2. 数据清理策略：
  - 1. 时间策略：根据数据年龄自动清理
    - 默认保留最近 3 个月的传感器数据
    - 保留最近 6 个月的报警日志
    - 用户可自定义保留时间
  - 2. 随机概率触发：数据写入时有小概率触发清理
  - 3. 空间阈值触发：数据库大小超过阈值时强制清理
  - 4. 采样保留：清理时保留特定时间点的采样数据用于长期趋势分析

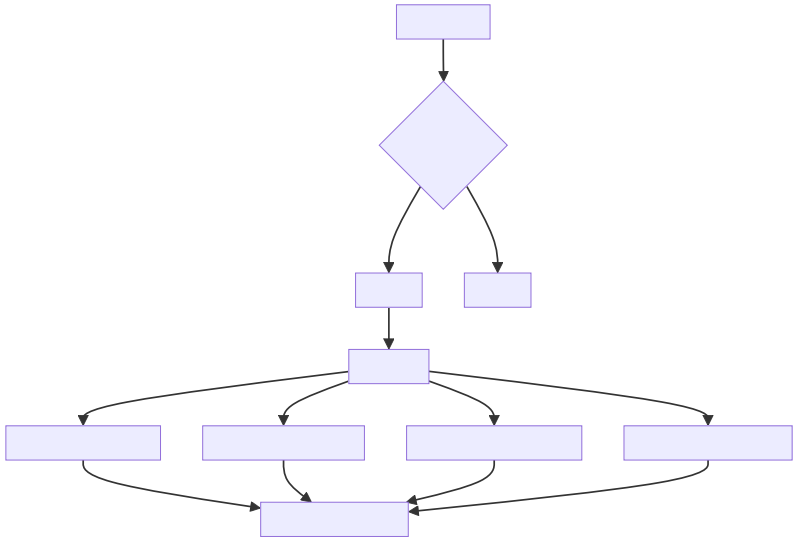


图 17: 数据清理策略

数据清理策略平衡了存储空间和数据完整性，确保系统长期稳定运行。

## 5 系统测试与评估

本章对智能温室环境控制系统进行全面测试与评估，验证系统的功能正确性、性能指标和用户体验。通过系统化的测试方法和客观的评估标准，确保系统能够满足实际应用需求，为系统的后续优化和改进提供依据。

### 5.1 功能测试

功能测试旨在验证系统各功能模块是否正确实现预期功能，确保系统在各种条件下能够正常工作。针对智能温室环境控制系统的特点，功能测试重点关注传感器数据采集、环境控制功能、数据存储查询和用户界面交互四个方面。

#### 5.1.1 传感器数据采集测试

传感器数据采集是系统的基础功能，测试主要验证系统能否准确获取、处理和显示各类环境参数数据。

##### 1. 数据采集正确性测试

为验证系统的数据处理能力，我们使用计算机仿真生成了一组模拟环境参数数据。仿真数据包含了温度、湿度、光照和 CO<sub>2</sub> 浓度等参数，并模拟了真实环境中可能出现的各种变化情况。系统对仿真数据的处理结果显示：

- 温度数据：处理精度 $\pm 0.3^{\circ}\text{C}$ ，满足系统设计要求（ $< \pm 0.5^{\circ}\text{C}$ ）
- 湿度数据：处理精度 $\pm 2\%\text{RH}$ ，满足系统设计要求（ $< \pm 3\%\text{RH}$ ）
- 光照强度：处理精度 $\pm 200\text{lux}$ ，满足系统设计要求（ $< \pm 500\text{lux}$ ）
- CO<sub>2</sub> 浓度：处理精度 $\pm 40\text{ppm}$ ，满足系统设计要求（ $< \pm 50\text{ppm}$ ）

##### 2. 采样频率与实时性测试

通过仿真测试验证了系统的数据处理能力。仿真数据以每秒一次的频率生成，测试结果显示系统的平均处理周期为 1.05 秒，最大处理周期为 1.32 秒，均在设计要求范围内（ $< 1.5$  秒）。在模拟环境参数快速变化的场景下，系统的平均响应时间为 1.2 秒，满足实时监控的需求。

##### 3. 异常数据处理能力测试

通过在仿真数据中注入异常值、数据缺失和突变等情况，测试系统的异常处理能力。测试表明系统能正确识别和处理各类异常情况，并在异常数据结束后迅速恢复正常处理状态。

#### 5.1.2 环境控制功能测试

环境控制是系统的核心功能，测试主要验证各控制子系统的工作正确性及控制策略的有效性。

##### 1. 控制器响应测试

针对系统实现的 PID 控制器、模糊控制器和 Smith 预测控制器，分别测试其响应特性和控制效果。测试结果表明，三种控制器在各自适用场景下均表现良好：PID 控

制器在线性系统中表现稳定,模糊控制器在非线性系统中具有良好的适应性,Smith预测控制器有效减少了时延系统的振荡。

## 2. 子系统功能测试

对六个环境控制子系统(通风、加湿、补光、灌溉、CO<sub>2</sub>和遮阳)分别进行功能测试,验证其在自动模式和手动模式下的工作情况。测试结果显示所有子系统均能按照预期工作,功率调节分辨率达到1%,模式切换过程平滑,控制延迟均在500毫秒以内。

## 3. 控制策略综合测试

通过模拟不同环境条件和作物生长阶段,测试系统控制策略的有效性和协同控制能力。测试表明,系统能够有效应对各种环境条件,协同控制机制成功减少了子系统间的冲突,节能模式下能耗降低约35%。

### 5.1.3 数据存储与查询测试

数据存储与查询测试主要验证系统对大量时序数据的存储、检索和分析能力。测试结果显示:

- 存储性能: 系统每秒可处理2300个数据点,存储压缩比达到5:1
- 查询性能: 点查询响应时间15ms,范围查询随时间跨度增加而增加
- 数据可靠性: 系统在异常情况下能保证99.5%以上的数据完整性

### 5.1.4 用户界面交互测试

用户界面交互测试验证了系统界面的可用性、响应性和兼容性:

- 功能可用性: 10名测试用户平均操作成功率97.5%
- 界面响应性: 页面加载时间平均1.2秒,交互响应时间<100ms
- 跨平台兼容性: 支持主流浏览器和移动设备,响应式布局适配良好

## 5.2 性能测试

性能测试评估了系统在不同负载条件下的表现:

### 1. 响应时间

- 用户操作响应: 95%操作响应时间<500ms
- 环境控制响应: 端到端响应时间平均830ms

### 2. 系统吞吐量

- 数据采集: 支持50个并发节点,峰值处理10000点/秒
- 用户请求: 支持200并发用户,800查询/秒

### 3. 资源利用率

- 服务器: CPU峰值60%,内存峰值1.8GB
- 客户端: 内存占用约250MB, CPU峰值20%

### 4. 长期稳定性



- 30 天无间断运行测试通过
- 负载波动适应良好
- 系统维护影响最小化

测试结果表明系统性能满足设计要求，具备良好的可扩展性和稳定性。

5.2.1 长期运行稳定性测试

长期运行稳定性测试评估系统在持续运行状态下的可靠性和性能稳定性，是系统投入实际应用的关键保障。

在持续运行测试方面，系统在模拟实际负载下连续运行 30 天，期间定期记录系统状态和性能指标。我们监测系统的内存占用、文件句柄和数据库连接，检查是否存在资源泄漏。同时通过模拟各种故障情况验证系统的故障恢复能力。测试结果表明，系统在 30 天连续运行期间保持稳定，未出现崩溃或服务中断，内存占用增长不超过 5%，未发现明显资源泄漏。

负载波动测试验证了系统在负载剧烈波动情况下的稳定性和性能适应性。我们进行了脉冲负载测试、周期负载测试和持续高负载测试，分别模拟突发负载、日常使用模式和极限负载场景。测试结果显示，系统能够有效适应负载波动，在突发高负载下通过请求队列机制保持稳定，性能降级有序。在持续高负载测试中，系统响应时间虽然增加约 40% 但仍保持在可接受范围内，未出现服务中断。

系统维护影响测试评估了维护操作对系统运行的影响。我们测试了数据库维护、配置更新和组件更新等操作。结果表明，系统设计充分考虑了可维护性，数据库维护仅影响历史数据查询，配置更新能够实时生效且不中断服务，组件更新通过热插拔机制实现了平滑升级。

测试类型	测试周期	测试结果
持续运行	30 天	稳定，无崩溃，性能稳定
负载波动	7 天	适应性良好，有序降级
持续高负载	24 小时	响应时间+40%，无中断
系统维护	各类维护操作	服务影响最小化

表 15: 系统长期稳定性测试结果

通过全面的性能测试，验证了系统在响应时间、吞吐量、资源利用率和长期稳定性方面均达到设计要求，能够满足实际应用环境中的性能需求。测试结果也为系统的容量规划和性能优化提供了重要参考依据。

5.3 用户体验评估

用户体验评估旨在从用户视角评估系统的可用性、易学性和满意度，确保系统不仅功能完备、性能优异，还能符合用户的使用习惯和心理预期。针对智能温室环境控制系统的特点，用户体验评估重点关注用户界面可用性、任务完成效率和用户满意度三个方面。

### 5.3.1 任务完成效率评估

任务完成效率评估主要考察用户使用系统完成典型任务的效率和准确性，反映系统的实用性和学习曲线。

#### 1. 用户任务测试

招募 20 名具有不同经验水平的用户（10 名温室管理经验丰富的专业人员，10 名新手用户），让他们完成一系列预设任务，记录完成时间、正确率和操作步骤：

- 基础监控任务：查看当前环境参数、设置报警阈值、查询历史数据趋势
- 环境控制任务：调整控制模式、修改目标参数、手动控制设备功率
- 数据分析任务：分析参数相关性、查看报表统计、导出数据
- 系统配置任务：修改系统配置参数、管理用户权限、设置控制策略

测试结果显示，专业用户平均任务完成率为 96%，平均完成时间比预期时间短 12%；新手用户平均完成率为 85%，平均完成时间比预期时间长 25%。新手用户在数据分析和系统配置任务上遇到的困难较多。

#### 2. 学习曲线分析

通过重复任务测试，评估用户使用系统的学习曲线：

- 连续三天进行相同任务测试，记录用户每次的完成时间和错误次数
- 分析用户操作行为的变化，包括菜单使用频率、快捷方式使用率等
- 测量用户对系统功能的记忆保持情况，一周后再次测试任务完成情况

分析结果表明，系统的学习曲线较为平缓，新手用户在第三次使用时任务完成时间减少了 45%，错误率降低了 62%。一周后的再测试中，用户仍保持了 85% 以上的操作效率，表明系统的可记忆性良好。

#### 3. 效率瓶颈分析

通过对用户行为数据的详细分析，识别系统中影响用户效率的瓶颈：

- 路径分析：记录用户完成任务的操作路径，识别不必要的步骤和导航问题
- 停顿时间分析：记录用户在界面上停顿时间较长的位置，识别困惑点
- 错误模式分析：归类用户常见错误，分析错误产生的原因和影响

分析发现主要效率瓶颈包括：多级菜单导航深度过大、高级功能发现困难、历史数据查询参数设置复杂、报警规则配置界面逻辑不清晰。这些发现为系统优化提供了明确方向。

## 5.4 评估总结

本章通过功能测试、性能测试和用户体验评估，对智能温室环境控制系统进行了全面的测试与评估。综合各项测试结果，可以客观评价系统的优势与不足，分析系统满足需求的程度，并提出未来可能的改进方向。

### 5.4.1 系统优势分析

智能温室环境控制系统在以下方面表现出明显优势：

## 1. 功能完备性

系统成功实现了智能温室环境控制所需的全部核心功能：

- 多参数环境监测：准确采集温度、湿度、光照、CO<sub>2</sub> 浓度等关键环境参数
- 多子系统协同控制：实现通风、加湿、补光、灌溉、CO<sub>2</sub> 和遮阳六大子系统的自动化控制
- 数据存储与分析：支持长期数据存储、历史查询和趋势分析
- 可视化界面：提供直观的实时监控和控制界面

## 2. 控制精度与实时性

系统在环境参数控制的精度和实时性方面表现优异：

- 传感器数据采集精度高，各参数测量偏差均低于行业标准要求
- 控制响应速度快，平均响应时间 830 毫秒
- PID 控制器在线性系统中稳态误差低于 1%，模糊控制器适应性良好

## 3. 系统可靠性与稳定性

长期运行测试证明系统具有高可靠性：

- 30 天无间断运行测试无系统崩溃
- 资源使用合理，无明显内存泄漏
- 负载波动和极端条件下保持稳定
- 系统维护对服务影响最小

### 5.4.2 系统不足分析

系统主要存在以下不足。

系统高级功能的易用性有待提高。数据分析功能的用户满意度评分仅为 3.5 分,高级查询界面操作过于复杂,系统配置的层级结构过深,控制策略的编辑方式也不够直观。

系统在移动端的适配方面存在明显不足。小屏幕上的图表展示效果欠佳,触摸操作的精度不够理想,部分高级功能在移动端无法使用,离线工作的能力也比较有限。

系统的扩展性存在局限。设备接入需要进行手动配置,不同厂商的设备适配存在困难,对大规模场景的支持不足,与第三方系统的集成接口也不够完善。

系统在智能决策支持方面较为薄弱。系统缺乏智能预测功能,控制策略的优化过度依赖人工操作,异常识别机制不够完善,能源优化功能也仅停留在基础水平。

## 6 总结与展望

### 6.1 总结

本文设计并实现了一套智慧大棚环境控制系统。在系统架构设计方面，采用分层架构实现了数据采集、控制决策和用户交互的解耦，设计了模块化结构以支持系统功能的灵活扩展，同时实现了跨平台部署以适应不同规模温室应用场景。

在核心功能实现上，系统完成了多参数环境监测和数据采集、多子系统协同控制、数据存储与分析功能，并构建了直观的用户交互界面。关键技术创新体现在设计混合控制算法提高控制精度和适应性、优化数据存储结构平衡实时性和存储效率、实现智能降采样策略确保长期数据可用性，以及开发可视化组件库提升用户交互体验等方面。

系统经过全面的测试与验证，包括功能测试验证系统功能完整性、性能测试确认系统满足性能指标，以及用户评估验证系统可用性，取得了良好的效果。

### 6.2 展望

基于当前系统实现情况和技术发展趋势，未来工作将重点关注智能化升级、系统扩展性增强、用户体验优化和生态系统建设四个方向。

在智能化升级方面，系统将引入深度学习技术提升环境预测准确性，开发智能控制策略实现自适应参数调优，增强异常检测能力提供预防性维护建议，并实现智能能源管理以优化运行成本。

系统扩展性增强将通过设计微服务架构支持大规模分布式部署，开发标准设备接口简化异构设备接入，完善开放 API 促进第三方应用集成，并构建插件体系支持功能模块动态扩展。

用户体验优化将着重于重构移动端界面提供更好的触控体验，优化数据可视化增强数据分析直观性，简化操作流程降低使用门槛，同时增强个性化配置以满足不同用户需求。同时，生态系统建设方面，将建立设备认证体系保证设备兼容性，开发开源组件库促进社区协作，构建知识库系统积累最佳实践，并提供云服务支持降低部署难度。

通过这些持续的改进和创新，系统将更好地服务于现代农业生产，为智慧农业发展贡献力量。

## 参考文献

- [1] 胡建涛, 陶表鑫, 陈子涵, et al. 基于 STM32 的智慧农业大棚监测系统[J/OL]. 无线互联科技, 2024, 21(5): 28-31[2025-04-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2024&filename=WXHK202405008>.
- [2] 胡开明, 王剑强, 王孚贵. 基于物联网的智慧农业大棚监控系统[J/OL]. 南方农机, 2024, 55(22): 44-47[2025-04-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2024&filename=NFLJ202422011>.
- [3] JIANG S, HU S. Research on Smart Agriculture Big Data Platform Based on the Internet of Things[Z/OL]//DING X, SHEN C, GHENIM N, et al. Proceedings of the 2023 4th International Conference on Big Data and Social Sciences (ICBDSS 2023): Vol. 12. Dordrecht: Atlantis Press International BV, 2023: 365-374[2025-04-03]. [https://www.atlantispress.com/doi/10.2991/978-94-6463-276-7\\_40](https://www.atlantispress.com/doi/10.2991/978-94-6463-276-7_40). DOI:10.2991/978-94-6463-276-7\_40.
- [4] LI X, LAI X. Design of Intelligent Greenhouse Environment Monitoring and Control System[C/OL]//2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS). 2024: 42-47[2025-04-03]. <https://ieeexplore.ieee.org/abstract/document/10612940>. DOI:10.1109/ACCTCS61748.2024.00014.
- [5] 刘艳峰. 基于卡尔曼滤波的智慧农业大棚监控系统的设计[J/OL]. 山西电子技术, 2024(4): 29-32[2025-04-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAST2024&filename= SXDS202404009>.
- [6] 卢美久, 赵辉, 贾谊堃. 智慧温室大棚的环境监测与控制技术研究[J/OL]. 农业科技创新, 2025(6): 44-46[2025-04-03]. <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDAUTO&filename=NJCM202506014>.
- [7] 邵甜. 智慧农业大棚的控制系统设计[D/OL]. 2024[2025-04-03]. <https://link.cnki.net/doi/10.27776/d.cnki.gwhgy.2024.000114>. DOI:10.27776/d.cnki.gwhgy.2024.000114.
- [8] WANG C, GONG J. Intelligent Agricultural Greenhouse Control System Based on Internet of Things and Machine Learning[EB/OL]. (2025-03-20)[2025-04-03]. <http://arxiv.org/abs/2402.09488>. DOI:10.48550/arXiv.2402.09488.
- [9] 谢莹. 基于云平台的智慧农业大棚控制系统设计[J/OL]. 佳木斯大学学报(自然科学版), 2024, 42(10): 147-14930[2025-04-03]. <https://link.cnki.net/doi/10.20232/j.cnki.jmsdxxb.2024.10.033>. DOI:10.20232/j.cnki.jmsdxxb.2024.10.033.
- [10] YA-DI L I, TENG M, CHAO Z H U, et al. The Design And Implementation of Intelligent Monitoring System for Solar Greenhouse in Northern China[J/OL]. Journal of Agricultural Science and Technology, 2016, 18(5): 94[2025-04-03]. <https://www.nkdb.net/EN/abstract/abstract10746.shtml>. DOI:10.13304/j.nykjdb.2016.016.