

设计题目5：锅炉蒸汽压力控制系统设计

MATLAB 部分文档

作者：周禄也

一、MATLAB仿真建模部分设计背景与意义

锅炉蒸汽压力控制系统是工业自动化领域的重要组成部分，其核心任务是让锅炉蒸汽压力始终保持在设定值附近，确保设备安全、稳定运行。设计高效、可靠的控制系统，不仅能提升生产效率，还能降低能耗、保障设备安全。

目前，PID控制器因其原理简单、性能稳定，在工业控制中应用极为广泛，尤其适用于锅炉蒸汽压力控制系统。在硬件条件一定的情况下，PID参数的整定往往决定了闭环系统的性能优劣。

本设计将利用MATLAB对锅炉蒸汽压力控制系统进行建模与仿真，主要包括以下几个方面：

- **系统建模**：建立锅炉蒸汽压力控制系统的数学模型。
- **控制器设计**：设计合适的PID控制器并进行参数整定。
- **系统仿真**：用MATLAB对设计好的控制系统进行仿真实验，验证性能。
- **结果分析**：分析仿真结果，评价系统的稳定性、响应速度和精度，并与理论分析对比。

二、设计思路与方法

2.1 系统建模

在设计锅炉蒸汽压力控制系统前，需对系统进行详细建模。根据锅炉的工作原理和控制要求，建立数学模型。常用方法有状态空间建模和传递函数建模。

1. 机理法

机理法基于锅炉系统的物理和化学原理，通过分析结构、能量和物质流动过程，建立数学模型。主要考虑：

- **质量守恒**：建立水和蒸汽的质量平衡方程，描述锅炉内水和蒸汽的变化。
- **能量守恒**：分析锅炉吸收热量、蒸发过程中的能量转换，建立能量平衡方程。
- **动量守恒**：考虑流体在锅炉内的流动特性，建立动量平衡方程。

实际锅炉系统较为复杂，通常需做如下简化：

- 忽略部分较小的时间常数和影响因素；
- 对非线性环节进行线性化处理；
- 采用集中参数法简化分布参数系统。

通过上述分析和假设，可推导出锅炉蒸汽压力控制系统的传递函数或状态空间模型，为后续控制器设计和仿真提供理论基础。

2. 实验法

实验法通过对实际系统进行试验，采集输入输出数据，建立系统模型。常用方法有：

- **阶跃响应法**：在系统稳态下施加阶跃输入，记录输出响应曲线。根据曲线特征参数（如上升时间、超调量、稳态值等）估算系统参数。适用于一阶、二阶等简单系统。
- **频率响应法**：对系统施加不同频率的正弦信号，测量输出的幅值和相位，绘制幅频和相频特性曲线（波特图），据此拟合系统传递函数。
- **系统辨识法**：采集系统在不同工况下的输入输出数据，利用最小二乘法等数学工具进行参数估计和模型拟合，适用于结构复杂或难以直接建模的系统。

实际工程中，常将机理法和实验法结合使用：

- 先用机理法建立系统的基本结构模型，保证物理合理性；
- 再通过实验法对模型参数进行修正和优化，提高准确性；
- 最后通过仿真和实际运行数据对模型进行验证和完善。

这种结合方法既能反映系统的物理本质，又能兼顾实际运行的动态特性。

2.2 控制器设计

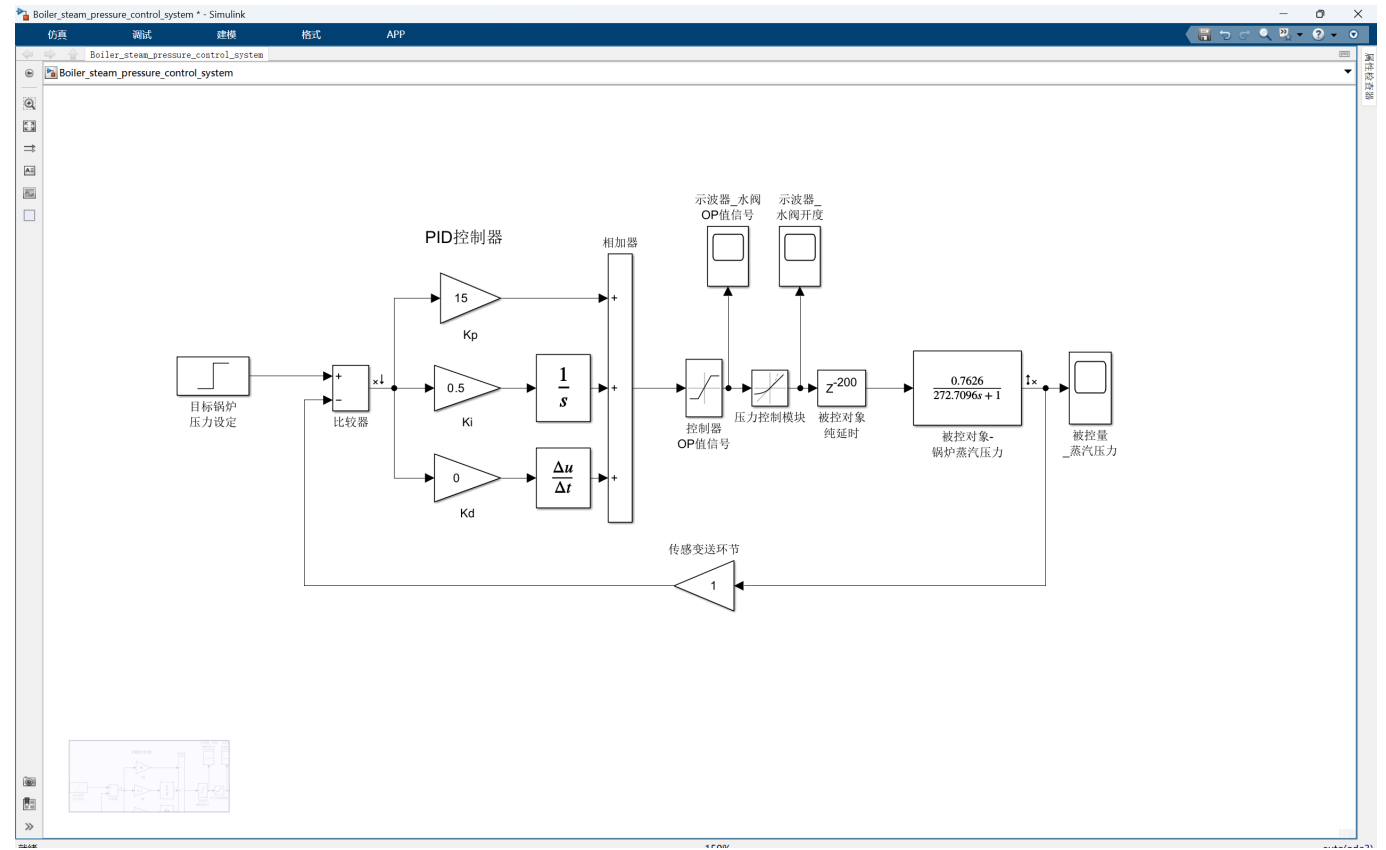
本次课程设计选用简单实用的PID数字控制器。PID控制器结构清晰，参数整定灵活，能够满足锅炉蒸汽压力控制系统对稳定性和响应速度的要求。

三、系统结构仿真与分析

3.1 系统总体结构

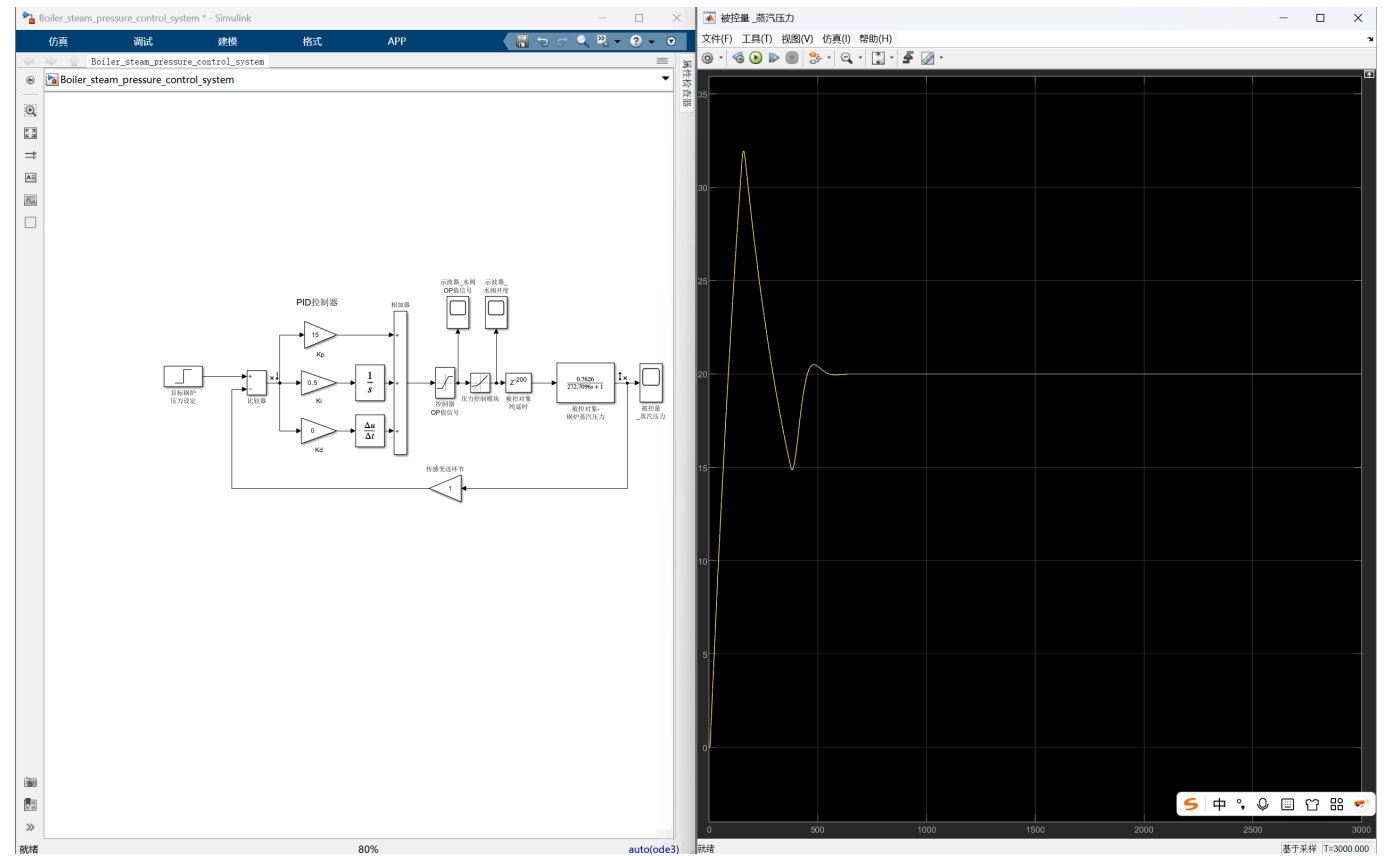
本系统采用单闭环控制结构，主要包括：比价器、PID数字控制器、执行器、被控对象、传感变送器。

锅炉蒸汽压力控制系统的总体结构如下图所示：



3.2 系统仿真

在MATLAB中，采用Simulink进行系统建模和仿真。下图为锅炉蒸汽压力控制系统的Simulink模型示例及仿真效果：



四、系统辨识简单小助手开发

4.0 功能展示

下图展示了“系统辨识简单小助手”的主要界面和功能布局：



4.1 设计目标

本模块开发了一个基于MATLAB GUI的“系统辨识简单小助手”，帮助用户通过实验数据点快速完成一阶系统的参数辨识与传递函数拟合。界面直观，操作简便，适合教学和工程初步建模。

4.2 功能需求

- 1. 数据输入功能
 - 支持用户逐点输入实验数据（时间T、液位高度Y）。
 - 提供数据点表格，实时显示已输入的数据。
 - 支持通过按钮或回车快捷键添加数据点。
- 2. 参数设定功能
 - 支持用户输入和设定水泵OP（操作点）参数。
 - OP参数可随时修改，界面有提示反馈。
- 3. 数据管理功能
 - 可清空输入框，便于连续输入。
 - 支持退出程序。
- 4. 系统辨识与拟合功能

- 至少输入3组数据后，支持一阶系统的参数拟合。
- 自动拟合一阶惯性环节模型： $y = K*(1-\exp(-t/\tau))$ 。
- 输出拟合得到的K、 τ 参数及传递函数表达式。
- 在界面和命令行输出结果，并绘制原始数据与拟合曲线对比图。

5. 用户交互与提示

- 对无效输入、数据不足等情况有友好提示。
- 操作成功后有反馈信息。

4.3 界面设计

- **主窗口**：包含输入区、参数区、数据表格区、结果显示区和操作按钮区。
- **输入区**：时间T、液位高度Y的输入框。
- **参数区**：OP输入框及设定按钮。
- **数据表格区**：显示已输入的所有数据点。
- **结果显示区**：显示拟合得到的传递函数。
- **操作按钮区**：添加点、计算、退出等按钮。

4.4 主要模块与流程

1. 初始化

- 初始化数据存储数组（X、Y）、OP参数、回车计数器。
- 创建GUI各控件并布局。

2. 数据输入与管理

- 用户输入T、Y后，点击“添加点”或按回车，数据自动存入数组并刷新表格。
- 输入框自动切换焦点，提升输入效率。

3. OP参数设定

- 用户输入OP值，点击“设定OP”按钮，系统校验输入有效性并保存。

4. 系统辨识与拟合

- 用户点击“计算”按钮，若数据点数 ≥ 3 ，则进行一阶系统拟合。
- 使用MATLAB `fit`函数，拟合模型 $y = K*(1-\exp(-t/\tau))$ 。
- 拟合结果（K、 τ 、传递函数）显示在界面和命令行。
- 自动绘制原始数据与拟合曲线对比图。

5. 用户提示与异常处理

- 对无效输入、数据不足等情况，界面有醒目提示。
- 操作成功后有确认信息。

4.5 关键代码说明

• 数据输入与表格刷新

```
X(end+1) = xVal;  
Y(end+1) = yVal;  
set(dataTable, 'Data', [X(:), Y(:)]);
```

- **OP参数设定**

```
OP = str2double(get(opEdit, 'String'));
```

- **一阶系统拟合**

```
fit_type = fitttype('K * (1 - exp(-x / tau))', ...);  
[fit_result, gof] = fit(X(:), Y(:), fit_type, 'StartPoint', initial_guess);
```

- **结果输出与绘图**

```
tfStr = sprintf('G(s) = %.2f / (%.2fs + 1)', K, tau);  
set(tfOutput, 'String', tfStr);  
plot(X, Y, 'bo', ...); plot(x_fit, y_fit, 'r-', ...);
```

4.6 设计亮点与改进建议

- 交互友好，适合初学者和工程人员快速上手。
 - 支持实验数据的可视化和一键拟合，极大提升辨识效率。
 - 可扩展性强，后续可增加多阶系统辨识、数据导入导出等功能。
-