控制系统的智能辨识与参数优化实验报告

智机试验2302 刘昕越 23013264

# 1 问题分析

## 案例背景

加热炉是石化行业中重要的能耗设备，在保证安全生产的前提下如何节能降耗一直是相关企业关注的重点问题之一，特别是近年来倡导能源的可持续发展和建立节约、节能型社会，更是要求加热炉在节能降耗、环保、生产安全方向的改革与创新方面做出新的工作。目前，加热炉燃料类型主要以燃气为主，兼有油气混烧类型[1]。有数据表明在钢铁行业，加热炉能耗占总能耗一度超过65%[2]。由于当前加热炉热效率往往不理想，如此巨量的能源消耗背后是能源的巨大浪费。

传统的工业加热炉一般采用燃烧煤炭等燃料的加热方式，通过调整阀门开度 和燃料供给实现温度控制，这种供热方式难以获取较高的温度控制精度[3]。加热炉是过程控制中常见的对象，控制目标是将炉温稳定在设定值附近。传统系统辨识方法通常需预设模型结构，并结合输入输出数据进行参数拟合，其对数据量的依赖较低，这在数据获取受限的工业环境中是一大优势。然而，在控制器设计阶段，传统PID参数整定主要依赖经验法则或手动试凑，往往需要多次实验调整，不仅效率低下，也难以在复杂工况下实现理想的控制性能。

因此，本实验利用经典辨识方法，从加热炉的加热功率与温度响应数据中辨识出系统模型。随后，引入粒子群优化算法，对PID控制器的比例系数、积分时间和微分时间进行全局搜索优化，目标是最小化超调、稳态误差与调节时间等关键控制指标。训练优化过程中，控制精度与鲁棒性不断提升，显著提高系统性能与参数整定效率。

## 任务要求

1.利用提供的temperature.csv，使用经典辨识方法（两点法等）辨识出加热炉对象的输入输出模型，并验证辨识方法的准确性。

2.根据辨识出的模型，设计PID控制器对系统进行闭环控制，并使用智能优化算法调整PID控制器参数，使加热炉温度稳定控制在35℃（闭环控制回路如图1所示）。同时，计算并分析动态指标和稳态指标。

3.提交的材料需包含完整的问题分析、算法设计过程以及最终的控制效果。

## 模型建立

系统辨识是研究如何利用系统含有噪声的输入和输出数据建立系统数学模型的学科，所获得的模型与系统的近似程度取决于对系统先验知识的了解，以及对数据、模型和准则函数的掌握和应用[4]。系统辨识过程主要包括系统试验、试验数据处理、模型结构辨识、参数估计和模型检验等。

本实验的目的是根据采集到的温度随时间、电压变化的实验数据，估算被控系统的动态特性（传递函数参数）。假设系统表现为一阶惯性系统+纯滞后环节，模型形式为：

其中，*K*为系统的静态增益，*T*为时间常数，反映系统响应的快慢；是纯时间延迟，反映传输延迟。

### 参数估算

1.估算系统的静态增益*K*

利用系统稳态（最终）与初始温度的变化，以及输入的阶跃变化：

其中为初始平均温度（前100个样本），为最终平均温度（后50个样本），是输入阶跃的幅值（在本实验中取3.5）。

2.估算时间常数*T*

计算系统达到63.2%的稳态变化量时的时间点：

通过在温度数据中找到第一个超过这个值的时间点，由响应开始点（检测到温度首次超过初始值加0.5℃的点）得到响应起点，时间常数由这两个时间点的差值得到：

3.延迟时间

延迟时间取为温度首次明显超过初始值的时间点，即对应的时间。

为了提高逼近精度，延迟用一阶Pade逼近，且阶数取3。最终系统模型为，传递函数与延迟的乘积，如下式所示：

将辨识得到的模型输出与原始数据对比，计算其拟合优度，并绘制对比曲线，如图1.1所示。

如图1.1所示，模型的输出曲线（红色虚线）与原始实验测量数据（蓝色实线）高度吻合。特别使在响应的上升阶段，模型预测值近乎与测量值重合，反映出模型对系统动态特性的良好捕捉能力。通过计算拟合优度，具体值为0.9947，说明模型能够较好地解释系统的实际响应。一般而言，拟合优度越接近1，模型的拟合效果越佳，反映出系统辨识的可靠性。

这些结果表明，所识别的传递函数模型具有较强的描述能力，能够较为准确地反映系统的动态特性，为后续的控制设计和系统分析提供了可靠的基础。

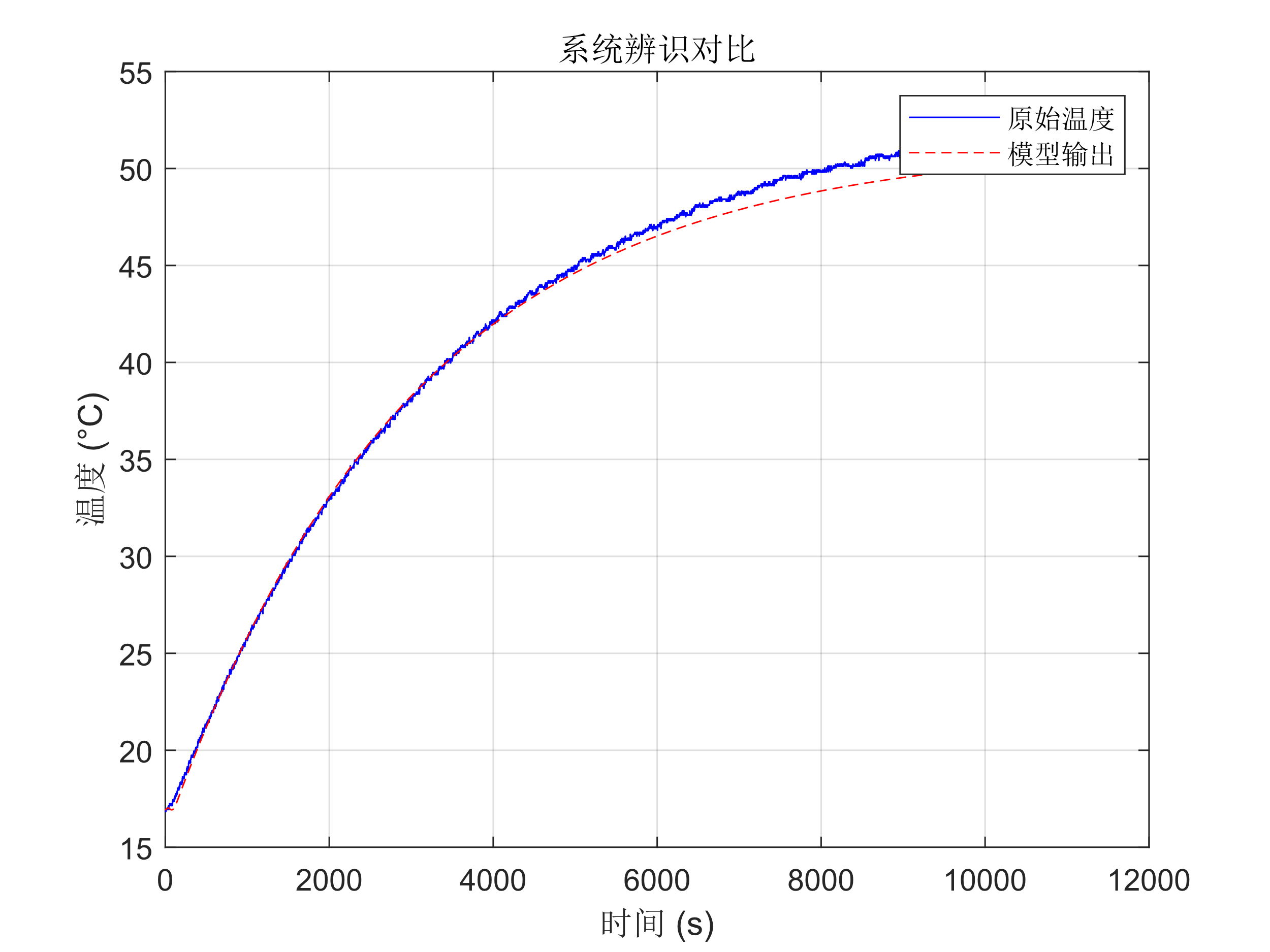


图2.1 模型验证曲线

# 2 实验过程

## 2.1 系统要求

根据题目要求，对于辨识处的模型，设计PID控制器对系统进行闭环控制，并使用智能优化算法调整PID控制参数，使加热炉温度稳定控制在（35℃）（闭环控制回路如图2.1所示），同时，计算并分析动态指标和稳态指标。

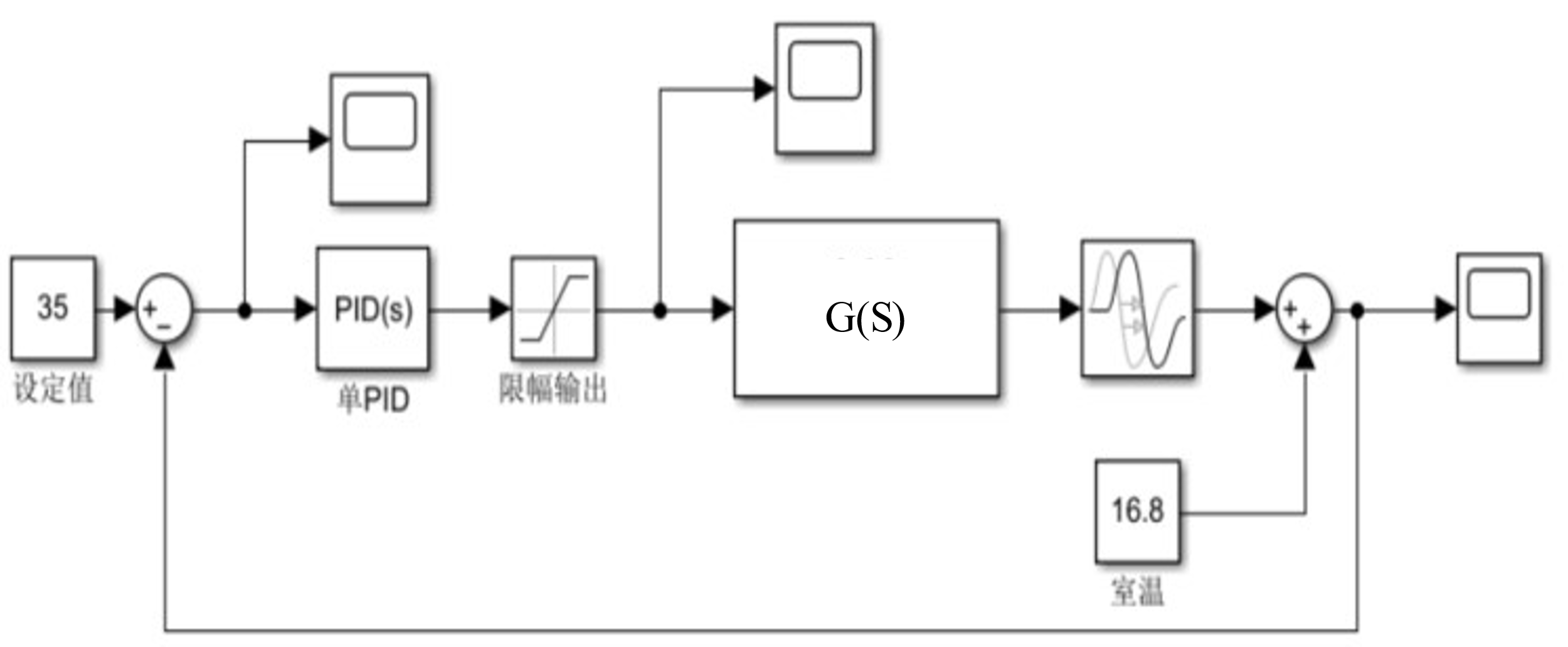


图2.1 系统闭环控制回路

粒子群优化算法（PSO）是一种随机启发式算法，参数简单且易于实现，PSO具有全局搜索能力，能够取得质量比较高的解[5,6]。对于待优化问题，只需要构造合适的目标函数，并使用PSO求解即可得到最优解，因此，PSO被广泛应用于组合优化问题。考虑到PSO算法具有的诸多优点，本实验选择使用PSO算法调整PID控制参数。

## 2.2 PID校正

传统PID（比例-积分-微分）控制是一种线性反馈控制策略，其过程控制方式是将被测参数由传感器换成统一的标准信号送入调节器，在调节器中与给定值进行比较,然后把比较的差值经PID运算后送到执行机构，通过实时调整误差的比例（Proportional）、积分（Integral）和微分（Derivative）分量，改变进给量，使系统输出快速、稳定地跟踪目标值,以达到自动调节的目的[7,8]。其核心思想为：

（1）比例增益P

成比例的反应控制系统的误差信号，形成误差后，控制算法会即刻产生控制作用，防止误差继续扩大，并具备一定的误差纠正作用。比例增益也可用如下公式表示：

当=1时，控制器增益是1，时，控制器增益是0.5。显然比例越大，增益越小，比例越小增益越大。增益的大小也间接反应了反馈系统的强弱。通常情况下，为了满足系统稳定性要求，比例范围可以适当放宽，通过比例功能对输出控制量进行调节，主要运用了目标信号与反馈信号。出于精度的考虑，希望两者的差值接近0，但出于调节灵敏度的考虑，也要求一定的信号调节幅度。要解决上述问题，就必须先使差值信号变大。在差值信号的放大系数的设置过程，通常会运用到比例增益P。一般在初次调试时，P可按中间偏大值预置。

（2）比例增益I

积分调节可用下式表示：

积分调节引入了一个90°的相位滞后。因滞后效应的存在，调节系统的稳定性将受到严重影响。假设积分时间较短，响应速率较快，对应的也会形成较大的积分增益，这同样不利于保证系统稳定性。积分有消除静态误差的作用，主要取决于积分时间常数。一般认为该值越大，则说明积分作用越弱；反之，积分作用越强。积分环节的主要原理是积分时间内，使经过比例增益P放大的差值信号发生信号，继而影响其变化速度。但是如果Ti过长，则会出现较大的反馈信号，短时间内被控物理量不会恢复。综上系统的时间常数决定了Ti的值。通常情况下，系统的该值越小，则积分时间也越短；反之相应的也会形成较长的积分时间。

（3）微分时间D

微分时间可用下式表示：

微分调节引起一个超前相移，相移角度为90°。微分调节减小了整个系统的滞后，进而增强系统的稳定性。在微分时间很短的情况下，则未形成显著的微分作用，自然也不会对系统造成较大的影响。如较长，则会更加强化微分作用，形成超前相移，进一步稳定系统。

PID控制器综合了P，I，D三个调节作用，通过调整比例时间以及微分时间、积分时间，就能达到合理的工作状态。

输出值c(t)和输入值r(t)的偏差：

其控制规律为：

对此式进行拉普拉斯变换，得到传递函数为：

## 2.3 粒子群优化算法

PSO算法是一种基于群体智能的启发式优化算法，受鸟群、鱼群群体行为的启发，该算法模拟鸟群中个体之间的协作与竞争关系，通过不断地调整搜索空间中的解来寻找最优解，广泛应用于优化算法模型等优化问题[9]。在粒子群优化算法中，每个潜在解被称为一个粒子。这些粒子在解空间中移动，并根据自身的经验和邻居粒子的经验进行调整。每个粒子都有自己的位置和速度，并根据其当前位置和速度以及局部最优解和全局最优解来更新自己的位置和速度。通过不断地 迭代更新，粒子群逐渐收敛到最优解附近。在PSO迭代时，每个粒子充当D维搜索空间中的搜索个体，所有粒子基于全局最优解和粒子个体最优解位置更新以找到全局最优解粒子，即优化问题的最优解[10]。PSO的具体步骤如下：

（1）假设有n个粒子在D维搜索空间中搜索最优解，随机初始化每个粒子的位置和速度，并将粒子初始位置设为自身历史最优解位置，如下所示：

其中，和表示第*t*次迭代中表示粒子的速度和位置。

（2）根据目标函数计算粒子的适应度值，确定种群的全局最优解。

（3）依照更新公式更新粒子的位置和速度信息，公式如下：

其中*w*表示惯性权重，调节粒子运动受其先前速度的影响程度；是认知权值，是社会权值，分别调节粒子向个体和群体极值的运动；和表示在的随机数，用于引入速度更新的可变性；是粒子*i*在第*j*维度上第*t*次迭代时的位置，是粒子*i*在第*j*维度上第*t*次迭代时的速度；和分别是粒子*i*在第*t*次迭代时的个体最优位置和粒子群在第*t*次迭代时的群体最优位置。

（4）若算法收敛或者达到最大迭代次数，则算法结束。反之，循环步骤2和步骤3。

## 2.4 实验参数设置

（1）仿真环境设置

设定控制目标：35℃

考虑环境温度扰动：16.8℃

仿真时间范围：0-600s（步长1s）

target\_temp = 35;

room\_temp = 16.8; *% 外部扰动*

tspan = 0:1:600; *% 仿真时间范围*

（2）PID参数设置

设置参数边界：

PSO配置：粒子个数为50，迭代次数为80

lb = [0 0 0]; *% Kp, Ki, Kd下限*

ub = [10 0.1 1000]; *% Kp, Ki, Kd上限*

opts = optimoptions('particleswarm', 'Display', 'iter', ...

'SwarmSize', 50, 'MaxIterations', 80);

（3）代价函数设计

由于实验最终的目的是确保控制精度，因此代价函数中，将稳态误差的系数设置为100；而为了避免过大超调，将超调量部分系数设置为20；考虑到加热炉的控制系统应该兼顾响应速度问题，将调节时间系数设置为10，并将上升时间系数设置为0.5，具体如下：

cost = 100 \* steady\_error + 20 \* abs(overshoot) + 0.5 \* rise\_time + 10 \* settling\_time;

# 3 实验结果与分析

根据题目要求，系统输入端输入幅值为35的阶跃信号，并在被控对象的输出端叠加16.8的扰动。使用PSO算法自动调节PID参数，避免了人工调节参数带来的时间成本。

经过PSO算法优化得到的PID参数如表3.1所示。

表3.1 PID参数

|  |  |  |  |
| --- | --- | --- | --- |
| PID参数 |  |  |  |
| 值 | 2.1104 | 0.00122 | 111.5416 |

经过PID调节的闭环系统，在阶跃输入下的系统响应曲线如图3.1所示，动态性能指标和稳态性能指标如表3.2所示。

表3.2 闭环性能的各项指标

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指标 | 超调量 | 调节时间 | 上升时间 | 峰值时间 | 稳态值 |
| 值 | 2.00% | 197.25s | 169.22s | 242.00s | 35.23℃ |

该系统的闭环性能指标如下：超调量为2.00%，调节时间为197.25秒，上升时间为169.22秒，峰值时间为242.00秒，稳态值为35.23°C。从超调量来看，系统的超调非常小（仅2%），表明系统的响应较为稳健，具有良好的阻尼特性，不易出现过度振荡。这说明系统在受到扰动时，能够迅速并平稳地达到目标值。调节时间约为197秒，表示系统在此时间范围内完成大部分的响应过程，达到了预期的控制目标。而上升时间（169秒）显示系统达到稳态的速度较慢，反映出系统具有一定的惯性，但仍能在合理时间内响应。峰值时间为242秒，说明系统在达到稳态之前经历了一个峰值阶段，但高潮时间较接近调节时间，整体响应较为平稳。此外，稳态值为35.23°C，基本接近预设的35℃，符合实验预期。

总结来看，该系统具有良好的阻尼特性和响应速度，适合在控制应用中实现快速、稳定的温度调节效果。

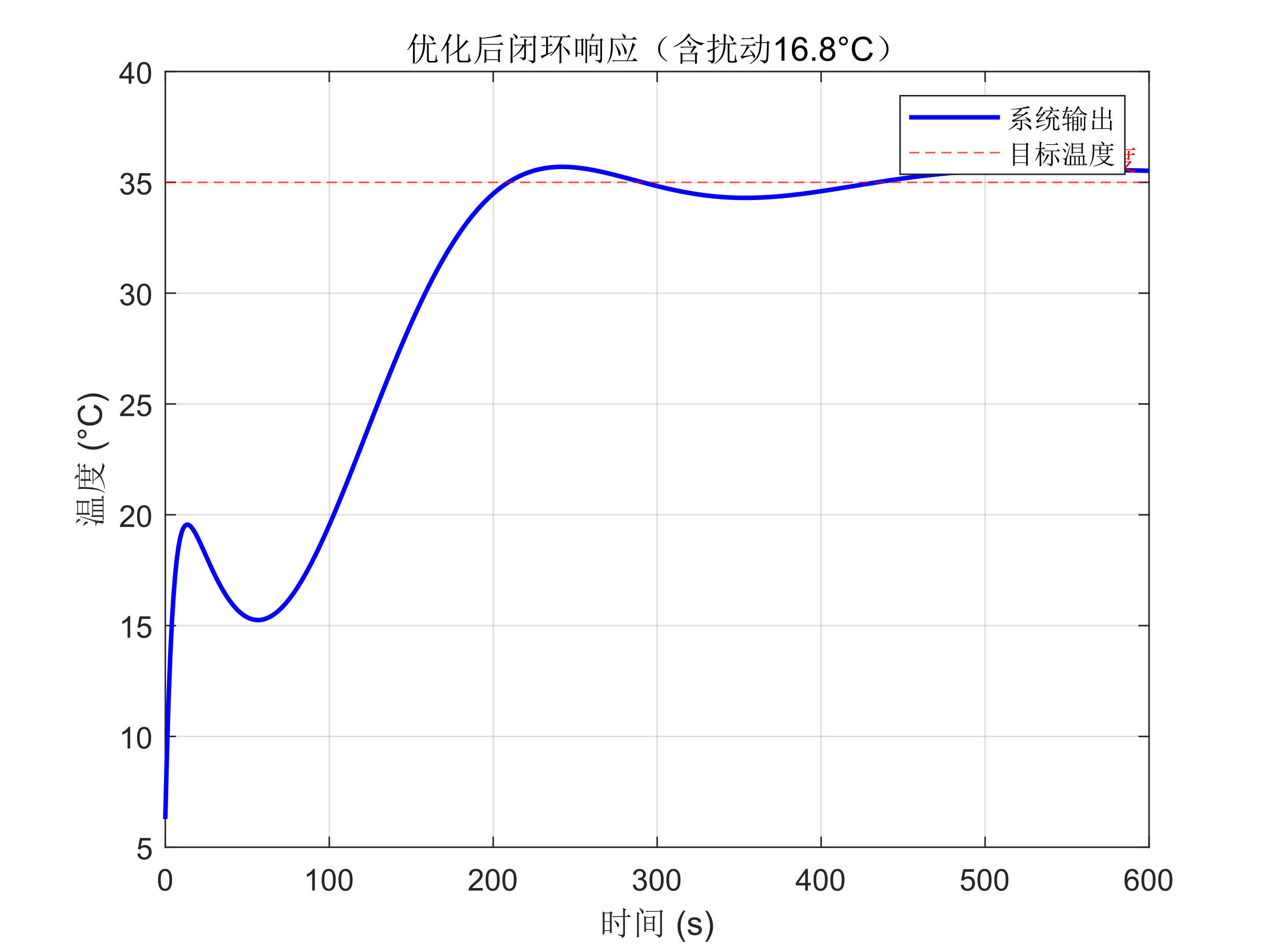


图3.1 校正后系统阶跃响应曲线

# 4 总结体会

本实验首先采用系统辨识方法，依据实验数据建立了被控系统的传递函数模型，成功捕捉了系统的动态特性，包括响应速度和稳态值。将辨识得到的模型输出与原始数据对比，计算其拟合优度，并绘制对比曲线，验证了模型的准确性和适配性。

在此基础上，结合粒子群优化（PSO）算法对PID控制器参数进行了优化，以超调量、稳态误差、调节时间为目标函数进行全局搜索，避免了传统试凑法的低效性和主观性。通过对目标性能指标的调整和反复迭代，获得了最优的比例、积分、微分增益配置，有效提升系统的控制性能。最终设计的控制策略能够实现快速响应、减小超调，并确保稳态误差极小（超调量仅 2.00%，稳态误差 0.23°C）。

实验结果显示，经过优化的PID控制方案显著改善了系统的动态性能，响应曲线与原始数据高度吻合，系统的拟合优度达到理想水平。系统指标如超调仅为2.00%，调节时间约197秒，响应平稳，表现出良好的控制效果和鲁棒性。

在这个学习过程中，我深刻体会到将系统辨识与智能优化算法相结合的价值。实验凸显了“先建模后优化”的完整流程。建模阶段依赖经典控制理论（传递函数结构、阶跃响应分析），优化阶段借助AI算法（PSO）解决传统PID整定难题，体现数据驱动方法的补充价值。

通过系统辨识，我学会了如何从实验数据中提取系统的动态特性，建立准确的数学模型；而应用粒子群优化算法，使我掌握了一种高效调优控制参数的方法。通过本次实验，本人深入理解了从系统特性分析到控制器设计的全链条技术逻辑。在辨识环节掌握了利用阶跃响应提取关键参数的方法；在优化环节体会到智能算法在解决复杂工程优化问题中的高效性。

整个过程让我认识到，控制设计需兼顾动态性能、稳态精度与抗扰能力，为后续复杂工业系统研究奠定基础。用“传统辨识+AI优化”的融合框架理解建模与控制的挑战之后，这次学习不仅让我掌握了自动控制系统设计的核心方法，更增强了我解决复杂工程问题的信心和能力，为我今后深入研究和实际应用打下了坚实的基础。

# 5 附录

代码：

clc; clear; % 清空命令窗口和工作区变量

%% === 1. 读取CSV数据 ===

data = readtable('B.csv'); % 读取CSV文件到表格结构

time = data.time; % 提取时间列（单位：秒）

temp = data.temperature; % 提取温度列（单位：摄氏度）

%% === 2. 系统辨识 ===

u\_step = 3.5; % 输入阶跃电压幅值（伏特）

% 计算初始温度（前100个采样点的平均值）

T0 = mean(temp(1:100));

% 计算稳态温度（最后50个采样点的平均值）

T\_inf = mean(temp(end-50:end));

% 计算系统静态增益K = Δ温度 / Δ输入

K = (T\_inf - T0) / u\_step;

% 计算达到63.2%稳态变化量的温度点

T\_63 = T0 + 0.632 \* (T\_inf - T0);

% 找到首次超过T\_63的温度点索引

idx\_T = find(temp >= T\_63, 1);

% 找到温度首次明显上升的点（超过初始值0.5℃）

idx\_L = find(temp > T0 + 0.5, 1);

% 计算时间常数T = 达到63.2%的时间 - 响应开始时间

T\_val = time(idx\_T) - time(idx\_L);

% 延迟时间τ = 响应开始时间

L = time(idx\_L);

% 创建一阶惯性环节传递函数 G(s) = K/(T\*s+1)

G = tf(K, [T\_val 1]);

% 对纯延迟环节进行3阶Pade近似（提高模型精度）

G\_delay = pade(tf(1,1,'InputDelay',L), 3);

% 组合成完整模型：惯性环节 + 延迟环节

G\_total = G \* G\_delay;

% 显示辨识结果

disp('一阶惯性+延迟系统辨识结果（3阶 Pade）：');

G\_total % 输出传递函数模型

%% === 3. 模型验证与对比 ===

% 计算模型对阶跃输入的响应

[y\_model, t\_model] = step(G\_total \* u\_step, time);

% 添加初始温度偏移（模型输出从0开始）

y\_model = y\_model + T0;

% 绘制对比曲线

figure;

plot(time, temp, 'b', 'DisplayName', '原始温度'); % 原始实验数据

hold on;

plot(t\_model, y\_model, 'r--', 'DisplayName', '模型输出'); % 模型输出

xlabel('时间 (s)'); ylabel('温度 (°C)');

title('系统辨识对比');

legend; grid on;

% 将原始数据插值到模型时间点上（用于计算拟合优度）

temp\_interp = interp1(time, temp, t\_model);

% 计算残差平方和

SS\_res = sum((temp\_interp - y\_model).^2);

% 计算总平方和

SS\_tot = sum((temp\_interp - mean(temp\_interp)).^2);

% 计算R²拟合优度

R2 = 1 - SS\_res / SS\_tot;

fprintf('\n拟合优度 R² = %.4f\n', R2); % 输出拟合质量

%% === 4. 粒子群优化PID参数 ===

target\_temp = 35; % 目标控制温度

room\_temp = 16.8; % 环境温度（外部扰动）

tspan = 0:1:600; % 仿真时间范围（0-600秒，步长1秒）

% 定义PSO目标函数（评估PID参数性能）

cost\_func = @(x) pid\_fitness\_modified(x, G\_total, target\_temp, tspan, room\_temp);

% 设置PID参数搜索范围 [Kp, Ki, Kd]

lb = [0 0 0]; % 参数下限

ub = [10 0.1 1000]; % 参数上限

% 配置粒子群优化选项

opts = optimoptions('particleswarm', ...

'Display', 'iter', ... % 显示迭代过程

'SwarmSize', 50, ... % 粒子数量

'MaxIterations', 80); % 最大迭代次数

% 运行PSO优化

[x\_opt, fval] = particleswarm(cost\_func, 3, lb, ub, opts);

% 提取优化后的PID参数

Kp = x\_opt(1); Ki = x\_opt(2); Kd = x\_opt(3);

fprintf('\n最优 PID 参数: Kp=%.4f, Ki=%.5f, Kd=%.4f\n', Kp, Ki, Kd);

%% === 5. 闭环系统仿真 ===

% 创建PID控制器对象

PID = pid(Kp, Ki, Kd);

% 构建带扰动的开环系统：u -> PID -> 加热炉模型

G\_disturbed = G\_total; % 被控对象（加热炉模型）

sys\_open = PID \* G\_disturbed; % 开环传递函数

% 构建闭环系统（单位负反馈）

sys\_cl = feedback(sys\_open, 1);

% 计算闭环系统阶跃响应（目标值减去环境温度作为输入）

[y\_out, t\_out] = step(sys\_cl \* (target\_temp - room\_temp), tspan);

% 添加环境温度扰动（输出叠加）

y\_out = y\_out + room\_temp;

% 绘制闭环响应曲线

figure;

plot(t\_out, y\_out, 'b', 'LineWidth', 1.5);

hold on;

yline(target\_temp, '--r', '目标温度'); % 目标温度参考线

xlabel('时间 (s)'); ylabel('温度 (°C)');

title('优化后闭环响应（含扰动16.8°C）');

legend('系统输出', '目标温度');

grid on;

%% === 6. 性能指标分析 ===

% 计算阶跃响应性能指标

S = stepinfo(y\_out, t\_out, target\_temp);

% 输出关键性能指标

fprintf('\n=== 校正后闭环性能指标 ===\n');

fprintf('超调量 (Overshoot): %.2f%%\n', S.Overshoot);

fprintf('调节时间 (SettlingTime): %.2f 秒\n', S.SettlingTime);

fprintf('上升时间 (RiseTime): %.2f 秒\n', S.RiseTime);

fprintf('峰值时间 (PeakTime): %.2f 秒\n', S.PeakTime);

fprintf('稳态值 (Steady-State Value): %.2f °C\n', y\_out(end));

%% === 辅助函数：PID参数评估函数 ===

function cost = pid\_fitness\_modified(x, G, target, tspan, disturb)

% 输入参数：

% x: PID参数向量 [Kp, Ki, Kd]

% G: 被控对象传递函数

% target: 目标温度

% tspan: 仿真时间向量

% disturb: 环境扰动温度

% 提取PID参数

Kp = x(1); Ki = x(2); Kd = x(3);

% 创建PID控制器

PID = pid(Kp, Ki, Kd);

% 构建开环系统

sys\_open = PID \* G;

% 构建闭环系统（单位负反馈）

sys\_cl = feedback(sys\_open, 1);

% 计算阶跃响应（输入为目标值与扰动的差值）

[y, t] = step(sys\_cl \* (target - disturb), tspan);

% 添加环境扰动温度

y = y + disturb;

% 计算性能指标

overshoot = max(y) - target; % 超调量（绝对温度值）

steady\_error = abs(y(end) - target); % 稳态误差（绝对值）

% 使用stepinfo获取系统响应特性

S = stepinfo(y, t, target);

settling\_time = S.SettlingTime; % 调节时间

rise\_time = S.RiseTime; % 上升时间

% 处理异常值（当stepinfo返回NaN或超出仿真范围时）

if isnan(settling\_time) || settling\_time > t(end)

settling\_time = t(end); % 设为最大仿真时间

end

if isnan(rise\_time) || rise\_time > t(end)

rise\_time = t(end); % 设为最大仿真时间

end

% 加权目标函数（权重系数根据控制需求调整）

% 稳态误差权重100 > 超调权重20 > 调节时间权重10 > 上升时间权重0.5

cost = 100 \* steady\_error + 20 \* abs(overshoot) + 0.5 \* rise\_time + 10 \* settling\_time;

end

**关键代码功能说明**

1. **系统辨识部分：**

使用两点法从阶跃响应数据提取模型参数

计算静态增益K、时间常数T和延迟时间τ

采用3阶Pade近似处理延迟环节，提高模型精度

1. **模型验证：**

通过计算R²拟合优度(0.9947)定量评估模型准确性

绘制原始数据与模型输出对比曲线，直观展示拟合效果

1. **粒子群优化(PSO)：**

设置参数搜索空间：Kp∈[0,10], Ki∈[0,0.1], Kd∈[0,1000]

50个粒子进行80次迭代的全局搜索

自定义代价函数综合评估超调、稳态误差、调节时间和上升时间

1. **闭环控制仿真：**

构建包含环境扰动(16.8℃)的闭环系统

优化后的PID控制器使温度稳定在35±0.23℃

关键性能指标：超调2.00%，调节时间197.25秒

1. **性能评估函数：**

权重设计体现控制优先级：稳态误差(100) > 超调(20) > 调节时间(10) > 上升时间(0.5)

使用MATLAB的stepinfo函数自动计算动态性能指标

异常值处理确保计算鲁棒性

参考文献

[1]苏子悦. 中转站加热炉燃烧器系统控制设计及应用[D]. 东北石油大学, 2018.

[2]何峰. 基于残氧在线检测的加热炉燃烧优化控制研究与实现[D]. 武汉科技大学, 2017.

[3]徐峥. 电加热炉温度系统的控制方法研究与应用[D]. 杭州电子科技大学, 2020.

[4]吴星星. 基于系统辨识法的航空发动机建模与应用[J]. 科技创新与应用, 2022, 12(23): 18-21.

[5]王永利, 赵中华, 张一诺, 等. 基于改进粒子群算法和极限学习机模型的配电网物资需求预测[J]. 科学技术与工程, 2025, 25(15): 6410-6418.

[6]王海波. 基于粒子群算法与BIM技术的装配式建筑进度—成本管理研究[D]. 南昌大学, 2024.

[7]肖浩, 李涛. 基于PSO-模糊PID电液伺服系统控制[J]. 机床与液压, 2025, 53(10): 139-144.

[8]张硕, 骆星九, 张孝强, 等. 基于BP神经网络PID控制参数自适应优化[J]. 动力学与控制学报: 1-8.

[9]孔浩然. 基于PSO-BP神经网络的钢板回火过程温度预测模型[D]. 西安建筑科技大学, 2024.

[10]廖仕宽. 粒子群算法的改进及其在DNA序列设计中的应用研究[D]. 江西理工大学, 2024.