基于粒子群算法的控制系统

PID参数优化设计

摘 要

本文研究了基于粒子群算法优化的PID控制器参数设计过程。传统常规PID的控制器参数（如临界比例法）整定常常不能满足大滞后对象的控制性能目标，本文以场景的加热炉温度控制为例为此进行研究。第一章首先介绍了本文的背景和课题研究意义，以及基于粒子群算法优化PID控制的可行性。第二章，阐述了粒子群算法思想的来源，粒子群算法的原理和过程，分析了粒子群算法的特点，本文采用带有惯性权重寻优方法。第三章，介绍了本文的控制对象——具有大滞后时间的加热炉温度控制，详细分析了该系统控制的特点，常规PID参数整定方法的难点等等，得出了加热炉的传递函数。第四章，详细介绍了采用PSO算法进行PID参数整定在Simulink仿真工具下面的过程，对比了传统工程下的PID参数整定效果，仿真实验表明，基于粒子群的算法比传统的在控制性能指标上有较大的优势。

关键词粒子群算法；参数优化；PID控制；SIMULINK

Optimal design of PID parameter of the control system based on Particle Swarm Optimization

Abstract

This paper studies the parameter design process of PID controller based on particle swarm optimization. Conventional PID controller parameters (such as the critical proportion method) often can not meet the control performance objectives of large lag objects, this paper takes the scene of heating furnace temperature control as an example to study this. The first chapter introduces the background and research significance of this paper, as well as the feasibility of optimizing PID control based on particle swarm optimization algorithm. The second chapter describes the origin of the particle swarm optimization algorithm, the principle and process of particle swarm optimization algorithm, analyzes the characteristics of particle swarm optimization algorithm, this paper uses the optimization method with inertia weight. The third chapter, introduces the control object of this paper -- the heating furnace temperature control with large lag time, analyzes in detail the characteristics of the system control, the difficulty of the conventional PID parameter setting method, and so on, obtains the transfer function of the heating furnace. In the fourth chapter, the process of PID parameter tuning using PSO algorithm in Simulink simulation tool is introduced in detail, and the PID parameter tuning effect under the traditional engineering is compared. The simulation experiment shows that the algorithm based on particle swarm optimization has a greater advantage than the traditional control performance index.

**Key word**: Particle Swarm Optimization; Parameter optimization; PID control; SIMULINK

目录

[摘 要 **I**](#_Toc14278)

[第 1 章 绪论 1](#_Toc71659990)

[1.1 研究背景和课题意义 1](#_Toc71659991)

[1.2 基本的PID参数优化方法 2](#_Toc71659992)

[1.3 常用的整定方法 3](#_Toc71659993)

[1.4 本文的主要工作与章节安排 4](#_Toc71659994)

[第 2 章 粒子群算法的介绍 5](#_Toc71659995)

[2.1 粒子群算法由来 5](#_Toc71659996)

[2.2算法原理 5](#_Toc71659997)

[2.3 算法流程 6](#_Toc71659998)

[2.4 全局模型与局部模型 7](#_Toc71659999)

[2.5 算法特点 8](#_Toc71660000)

[2.6 带惯性权重的粒子群算法 8](#_Toc71660001)

[第 3 章 用粒子群方法优化PID参数 9](#_Toc71660002)

[3.1 PID控制原理 9](#_Toc71660003)

[3.2 PID控制的特点 10](#_Toc71660004)

[3.3 优化设计简介 10](#_Toc71660005)

[3.4 目标函数选取 11](#_Toc71660006)

[3.5 大迟滞系统 12](#_Toc71660007)

[3.6 加热炉温度控制简介 14](#_Toc71660008)

[3.7 加热炉系统特性分析 15](#_Toc71660009)

[3.8 加热炉的传递函数 15](#_Toc71660010)

[第 4 章 系统仿真研究 16](#_Toc71660011)

[4.1 工程上的参数整定 16](#_Toc71660012)

[4.2 粒子群算法参数整定 18](#_Toc71660013)

[4.3 结果比较 20](#_Toc71660014)

[总结 21](#_Toc71660015)

[致 谢 22](#_Toc71660016)

[参考文献 22](#_Toc71660017)

[附录 A（程序清单） 23](#_Toc71660018)

第 1 章 绪论

1.1 研究背景和课题意义

自上世纪以来，自动化控制技术飞速发展，并且在如采矿、冶金、机器人等领域取得了惊人的成绩[1-4]。然而，在当下自动控制领域中，多数情况下，处于成本考虑，PID控制器仍然占据了很大一部分市场。从PID控制器优点来看，其无需知道受控对象的具体模型，而只需要调节三个参数即可完成反馈控制，而且还具有结构简单、鲁棒性不俗等优点。对于PID来说，其核心问题是寻找适合的控制器参数，在实际情况中，需要不断调试整定上面三个参数以达到满意的控制性能指标要求。

在过程控制领域中，控制对象多数是复杂的非线性、高阶、带时滞的[5]，这就对PID控制器带来巨大的挑战，往往其参数整定的过程是困难且复杂的。目前来看，设计PID控制器方法主要有以下几种：第一，根据对控制对象的经验或者试凑法来整定参数；第二：对受控对象进行建模从而针对性的根据目标控制性能进行PID参数的优化；第三，根据极点配置方法对PID参数进行调整。

近年来，研究人员通过对自然界的演化规律的学习，设计了多种优化算法，比如利用生物进化理论的遗传算法优化，应用在PID控制器领域，遗传算法可以通过自身的交叉、选择、变异等特性在特定PID参数内进行优化来达到好的性能。此外，以蚁群、鱼群和鸟类为启发而发现的群体智能算法也扮演了重要的角色。群体智能算法使用群体的随机搜索算法，通过定义群体行为和个体行为，使得群体有种群演化与行为指向性，来近似求解不能微分求导得到的优化问题。

对于控制来说，控制器的三个参数的整定问题是工业中经常遇到，可以归结为三参数的组合优化问题。为了解决这种类似的优化问题，人们提出了许多优化算法，例如比较著名的有神经网络算法、蚁群算法和遗传算法等。参数的优化整定主要包括两个方面来考虑：第一是要寻求全局最小值点，第二是要求要较好的收敛速度。神经网络参数整定目前在PID控制领域也有应用，优点是可以根据梯度下降的方向往局部最小值优化，进而得到较好的控制，但缺点是容易陷入局部最小值，且解释性不好；而遗传算法需要进行编码与解码的过程设计，有些情况下这些是非常困难的，不容易并行处理，计算量很大。

粒子群算法没有遗传算法的编码解码设计过程、交叉与变异的复杂思想，无需梯度信息，运行效率较快，容易实现，收敛速度快等特点，基于PSO的粒子群算法整定优化是一种更加简单、实用新型的整定方法，大大提高了PID三个参数的优化水平，优化了的控制系统性能指标得到了显著提升，在工业领域潜在价值巨大。

1.2 基本的PID参数优化方法

在现代控制领域，PID参数整定优化方法有很多，比如梯度下降法、遗传算法、蚁群算法等。梯度下降算法是以梯度为基础的无约束最优化问题的数值计算法，它的基本思想是定义一目标函数，求出其梯度，在其负梯度方向(最速下降方向)作为每步迭代的搜索方向，逐步逼近目标的极小值点；遗传算法来源于自然界生物进化优胜劣汰的思想，是模拟了自然选择和遗传学方法的生物进化过程的计算模型。遗传算法其核心思想是：先初始化一个种群，其经过设定数目的个体经过基金进行编码(个体表示为所求问题的解决方法)，然后定义种群的适应度函数，按照生物进化中的适者生存的原理，经过选择、交叉和变异进行逐代演化，产生出越来越好的个体（解决方法）。这样经过数次的演化，最终将使得最后的种群更加适应环境，种群中的个体更加适应目标的适应度函数，最后，问题的最优解就是最优种群中的最优个体经过解码后的值；蚁群算法是受自然界人们对蚁群集体行为研究启发而提出的基于种群的启发式全局进化算法。蚁群算法中核心思路如下：①最初，蚂蚁从蚁巢出发寻找食物，食物找到后在从食物源返回的路上释放出信息素；②别的来觅食的蚂蚁会跟随之前蚂蚁留下的信息素痕迹找到食物源。信息素是按一定的比例释放，路径越短，其释放的信息素越多，浓度也越高；信息素浓度越高，吸引的蚂蚁越多；吸引的蚂蚁越多，遗留下的信息素也越多。③最后当所有蚂蚁集中到信息素浓度最高的一条路径上时，这条路径就是从蚁巢到食物源的最短路径。

尽管学者们针对最优化问题提出了多种方法，但随之而来的是在工业控制和科学领域，大多不能接受太久的求解问题的时间，且工业过程控制中问题的非线性、多峰性且不可导更加给研究人员带来巨大的挑战。因此，发展高效率、高可用性、收敛快的优化算法总是必要的。

1.3 常用的整定方法

在工业领域过程控制系统中常用的PID参数整定方法有经验法、衰减曲线法、临界比例度法、反应曲线法等。衰减曲线法调整控制器器参数的方法是：先用纯比例控制下，即令为，为0，目标是让响应曲线按，得到衰减振荡过度过程曲线，若曲线衰减大于 ，应将朝较小比例方向调整；若小于，应让朝较大比例带方向整定。记下的比例带，并在所得曲线上求衰减时的调节周期，计算，，。

临界比例度法本质是考虑通过试验找到等幅振荡的过渡过程，得到临界比例度和等幅振荡周期。当操作输入变量作阶跃变化时，受控变量随时间变化的曲线称为反应曲线。对有自衡的非振荡过程，广义对象传递函数可用表示[6]。K，和T可用图解法得出，控制器参数整定则依据广义对象的K，和T来确定。

不同的场景下对不同控制性能有不同的侧重，是权衡了鲁棒性和快速性得出的结果。如果强调快速跟踪系统，则要求调节时间短，而强调鲁棒性好稳定的系统则要求超调量小。最终控制系统的系统指标好坏除了控制对象的的内在因素外，还有本身设计的控制器的参数整定。PID控制器的参数整定问题对于当下工业过程控制对象来说并不简单，参数在整定时要考虑控制对象的各种特性，还有会影响系统运行的随机干扰噪声等。而且，控制器参数本身的调整也会对系统的特性产生重大影响。总结来看，PID调节器的参数对受控对象的控制性能的影响主要在：

1. 比例带：比例带越小，上升时间响应减小，衰减比*S*减小，稳定度下降。在工程上，比例带常用P来描述。
2. 微分：微分的大小由微分时间来决定。越大，越能克服系统的滞后性和，能够缩短调节时间，起到降低超调量的作用。
3. 积分：积分是由积分时间来体现。越小，消除静差越快，但可能会引起系统稳定度下降，甚至导致系统震荡的发生。

实现自整定的PID控制，需要对被控制对象有先验知识，再选择相应的自整定计算方法来完成。总体来说，可将PID控制参数自整定方法分成两类：模型法和经验规则法。基于模型法的PID控制器参数自整定，是通过建立被控对象的数学模型，在此基础上对系统内部状态进行分析后计算的PID整定参数。而基于经验规则的PID控制器参数整定方法，则是运用对被控对象经验上的表征总结，根据专家经验使得系统在响应曲线上得到更好的性能 [4]。

控制理论经过多年发展，出现了许多基于现代控制理论的先进控制方法，但是工业实践中出于成本、受控对象难以建模等考虑，实际中控制仍然是用PID控制为主（占比超过90%），PID参数的整定尽管已经有了一些有效的经验方法，但是手动整定PID参数仍然是一件费时费力的事情，因此学术界研究出了很多自整定PID算法[5-8]。

为了提高传统PID参数整定的鲁棒性，学者结合多种启发式搜索算法，如遗传算法，粒子群算法等，此外模糊专家控制与PID控制结合的模糊PID也广泛地应用于PID控制器参数自整定中[7]。

1.4 本文的主要工作与章节安排

本文采用进化算法中的粒子群算法对PID参数进行寻优，主要工作如下：①选择控制系统的目标函数，②针对步骤一的控制系统，搭建基本的PID控制器模型③应用工程上的临界比例度法，作为基本的整定方法粗略确定初始PID控制参数，在此基础上进行优化④在Simulink仿真中应用粒子群算法对控制系统进行逐步仿真，对结果进行分析，得到最好PID控制参数。⑤应用优化后的PID控制器参数，得出该系统的响应曲线，对比之前的基础PID参数，发现它的性能指标，都比原来的曲线有了很大的改进。

第 2 章 粒子群算法的介绍

2.1 粒子群算法由来

生物体在自然界中存在一定的群体化行为，这是粒子群算法起初的最核心思想来源，例如自然界鸟群的迁徙与觅食、鱼群的聚集等等。1987年，有学者把鸟类的这些特性总结起来，以数学化模型化的思想提出了鸟群聚集模型，模型的约束条件如下：

(1)鸟群的个体要避免与相邻的碰撞；

(2)鸟群的个体要匹配相邻个体的速度；

(3)个体都要朝着鸟群中心聚集。

遵循以上三条规则，就可以仿真出自然中鸟类的飞行模型。当时收到学术界这种思想的启发，1955年，粒子群算法提出。粒子群算法核心思想是团体协作与信息源共享，其模拟了鸟类觅食行为的特性，把鸟类比作粒子，多个粒子在一个平面上搜寻目标，当自己离目标越近时，会把自己的信息共享给其他个体，然后其他个体会逐步靠近，通过这种行为找到最后的最优解。

2.2算法原理

在K维的空间中，有n个无质量微粒子组成的粒子群，每个微粒子是一个K维向量，其空间位置表示为xi =(xi1，xi2，…，xiK)，i=1，2，…n，而它的K维空间位置是提出问题中的潜在解，然后带入目标优化函数算出各个粒子对应的适应值，根据适应度值评价xi的好坏。第i个粒子的速度也是一个K维的向量，被记为vi=(vi1，vi2，…，viK)；第i个粒子迭代周期中最优适应值的位置称为该粒子的历史最好位置，记为pi=(pi1，pi2，…，piK)；整粒子群所达到的最好位置称为全局历史最好位置，记为pg=(pg1，pg2，…，pgK)，个体更新迭代方程为：

 (2.1)

 (2.2)

其中：下标j表示是第j维，下标i表示第i个微粒子，c1和c2为加速常量，通常为0到2之间，下标t表示第t代，r1 和r2 为均匀分布的两个独立随机函数[7]。

公式(2.1)为微粒子更新公式，分别由微粒子上一次的速度、个体自身认为的最好矢量项、种群最优矢量项组成。该表达式充分体现了粒子群的思想，即包括当前状态、个体经验、群体经验来决定下一次该如何前进，这也是粒子群优化的核心思想与标准表现形式。

2.3 算法流程

基本粒子群算法的流程如下：

（1）粒子群超参数以及随机解初始化。

（2）计算各粒子的适应度值。

（3）如果当前时间下的粒子适应值比之前所有的都号，则更新最优值。

（4）迭代每个例子，如果当前粒子比群里最好位置适应值更优，则其作为群体最优。

（5）依据表达式(2.1)和表达式(2.2)对粒子的速度、位置进行更新。

（6） 如果满足全局适应值足够优或运行达到最大迭代次数），则结束，否则转到步骤(2)。

粒子群优化算法的流程如图2.1所示

输出结果

依据式(2.2)更新粒子的位置

依据式(2.1)更新粒子的速度

求出群体的全局最优值

更新各个粒子的个体最优值

计算各个粒子的适应值

初始化粒子的速度和位置

是否满足结束条件

是

否

开 始

图2.1 基本粒子群算法流程图

2.4 全局模型与局部模型

在2.2节中，粒子的速度和位置受自身最优状态和全局最优状态的影响，依据这种更新式的粒子群被称为全局粒子群算法，如图2.2所示。另与之相对应的是局部粒子群算法，其中的粒子更新状态不受全局最优状态的影响的，只受到自身最优和相邻粒子中的局部最优状态的影响的，其拓扑如图2.3所示。局部粒子群算法速度更新公式为式为：

(2.3)

其中，pij为相邻粒子的局部最优值。

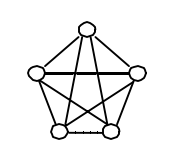
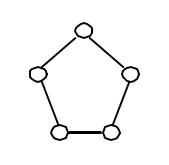
 

图2.2 全局最优模型 图2.3 局部最优模型

2.5 算法特点

粒子群算法原理简单易懂，超参数较少，容易实现，且收敛速度不错，因此在参数优化方面广泛应用，其计算量较小、占用计算资源少。目前，粒子群算法能很好地进行多种全局优化已成为业内的共识。PSO算法容易陷入局部最优值，导致后面向全局最优收敛变慢等缺点。

2.6 带惯性权重的粒子群算法

基本的粒子群算法容易陷入局部最优值，导致后面全局最优解收敛过慢，因此，为了提高粒子群的全局与局部搜索能力，学者们提出了带惯性权重的粒子群算法，更新参数公式为：

 (2.4)

 (2.5)

式中w是更新的惯性权重，表示本次更新保留原来的状态的量比。w越大，算法的全局收敛越快，局部目标收敛越弱；w越小，局部收敛越快，全局收敛越弱。适合的权重w可以加快算法的进行，减小算法收敛时间。

第 3 章 用粒子群方法优化PID参数

PID控制器作为最先发展起来的控制理论方法，原理结构简单，在当下的工业控制中占据最为主导的地位，其是将目标值的误差的比例(P)、积分(I)和微分(D)值通过线性组合的方式构成输出控制量，输出控制量对被控对象进行控制的方法。随着现代电子计算机的普及，数字PID控制在实践生产过程中已成为一种最常用的控制方法，在冶金、机器人、化工、机床等诸多领域中应用广泛。

3.1 PID控制原理

PID控制器由三个参数组成，整个控制闭环中，给定目标值为r(t)，目标偏差为e(t)，pid输出为u(t)，被控对象输出为y(t)，具体如图3.1所示。

比例

积分

微分

被控对象

r(t)

e(t)

u(t)

y(t)

+

+

+

–

图3.1 PID控制系统原理框图

PID是基于目标值误差的控制方法，误差计算方法为：

 (3.1)

PID的控制输出为：

 (3.2)

传递函数的PID形式为：

 (3.3)

其中，被称为比例系数，为积分时间常数，为微分时间常数。PID各部分作用如下：

1、比例环节：偏差信号e(t)与比例系数相乘，当偏差产生，比例环节将立即产生控制作用，以减小偏差；

2、积分环节：偏差信号e(t)的积分与积分系数相乘，用来消除稳态后的静差，提高系统的精确性；

3、微分环节：偏差信号e(t)导数与微分系数相乘，以反应偏差的变化趋势，并能在误差信号变得太大之前，在系统中提前引入修正值，从而加快系统的控制速度，达到减少调节时间目的。

3.2 PID控制的特点

PID控制器自提出以来，一直被广泛应用在工业过程控制领域，其关键在于其原理简单易懂、控制效果优良、无需知道受控对象数学模型经过多次试凑即可得出良好的曲线。

但是现代工业的飞速发展对机器的精度要求提出了很高的要求，工业生产中的控制对象大多是非线性、带滞后环节、高阶的甚至是时变的，因而对建立精细的数学控制模型非常艰难，然而常规PID很难跟踪理想的目标，效果欠佳。此外，在实际生产的实践中，常规PID控制器的整定过程往往是繁杂的、耗时耗力的，而且不好的PID参数容易出现，控制结果性能欠佳的情况。

3.3 优化设计简介

所谓优化设计，就是对已知问题的寻优的过程。在本论文中，对于一般受控对象而言，经过基本试凑法整定的PID参数是不够好的，其对应的曲线响应往往是欠阻尼或者过阻尼的，因而需要设计一种优化方法，来动态的找出一组足够优秀的PID参数，这就是一种寻优过程。

优化需要建立相应的数学模型，其可由组成变量、目标函数和约束条件三部分组成。

（1） 组成变量：根据优化目标的不同，设计不同的组成变量，这些变量方便对实际真实物理模型的表示，进而方便带入目标函数进行优化。

（2） 目标函数：目标函数表征达到目标的物理上的数学模型，优良的设计出目标函数可以直观的表征实际对象达到最优化后，该目标函数会达到最小值。

（3） 设计约束：有时候可能目标函数的变量，会有一定的范围约束，目标函数的变量必须在这些范围内取值，这样得到的最优值点才是合法的。

参数的优化方法有多种类型，如果按组成变量的数量划分，可将优化分为单变量优化和多变量优化；如果按约束条件有无划分，又可分为无约束优化和有约束优化；如果按目标函数数量划分，又有单目标优化和多目标优化两类。

3.4 目标函数选取

在参数最优化的问题中，需要规定目标函数的性能评价指标，被称为目标函数。选择不同的目标函数目的是反应评价系统性能指标的不同维度，最好得到的评价函数是容易计算的。选择不同的评价的目标函数，对于同一控制系统来说，优化后得到的参数可能会有所不同。

几种常用的目标函数如下：

（1）误差平方的积分型，其表达式为

 (3.4)

其中e(t)=r(t)-y(t)表示响应与期望值的误差。一般要求e(t)越小越好，即要求控制系统的输出响应y(t)尽可能的接近输入r(t)。由于在到调节稳态的过程中e(t)可正可负，因此取误差的平方进行积分来抵消符号的影响。这种目标函数简单好用，计算效率很高，但是，有时候不同时期的误差对于系统的重要程度是不完全相同的，有时不能特别好的反应真实对系统的性能指标的要求。

（2）时间乘以平方误差型，表达式为

 (3.5)

该式子在误差的平方上乘以了时间t再积分，相当于给误差加上了时间权重，这样是为了减少初始阶段大误差对系统的影响，而更看重后期出现的误差。

（3）误差绝对值积分型，表达式为

 或者为  (3.6)

该式子除了加了误差的绝对值以消除符号影响外，又另外加了时间t的权重，这样过度过程中后期出现的误差也基本上能消除。本文选择的优化目标函数的表达式取。

3.5 大迟滞系统

在工业生产制造过程中，被控对象常常有不同程度的纯延时，例如在热交换器过程中，传感器测量加热物料的出口温度，而控制量是加热介质，当改变加热介质的流量后，一段演示时间后才回对物料的出口温度产生影响。此外，如水箱系统、加热管、反应炉等多个设备都存在比较大的滞后时间。由于这些控制对象的纯滞后的存在，使得被调量不能及时反映系统所受的扰动，即使传感器的输出信号输入到控制器中，控制机构接受到控制器信号后立即动作，也需要经过时间以后才到达被调量。因此具有纯滞后的系统认为是非常困难的系统。工业生成种，一般按经验认为与系统时间常数之比大于0.3，则认为该系统是大滞后的系统。当/T逐步增加，系统相位滞后加剧，使得系统更加难以控制，有时甚至会因为系统超调量过于严重而出现停产事故。极端时候则可能引起系统不稳定，系统超出安全警戒，从而危及设备及人身安全。因此大迟滞系统一直被受人们的关注，成为重要的课题之一。

在大迟滞系统中，常常采用补偿方法来优化控制，其按照控制对象特性设计出一种模型加入到反馈控制系统中，来补偿系统过程控制的动态特性。常用的有史密斯预估器，其由Smith先生与1959年提出。其主要原理是预先估计出控制的动态模型，然后将Smith预估器并联在被控对象上，让其对过程中的纯滞后特性进行补偿，预期是将被延迟的时间的被控量提前送入到控制器中[8]，让其能提前做出反应，这样就通过补偿器的方法来消除滞后带来的影响。在实际生产中，如果在模型匹配的条件下能够有效减少超调、可以获得比较好的控制效果。

带有Smith补偿器原理如图3.2所示。















+

–

图3.2 带有Smith补偿器的原理示意图

图中——PID调节器；

——被控对象的传递函数模型，其中为不包括纯滞后时间的对象模型；

——Smith预估补偿器。

在未进行Smith预估补偿情况下，系统闭环传递函数为

 (3.7)

故其闭环特征方程式为

=0 (3.8)

由于在闭环方程中带有纯时间滞后项，会在闭环系统中引入了相角滞后，增加了系统的控制难度。引入Smith预估补偿器后是使调节器控制的等效对象中能消除纯滞后部分[9]。图3.2的数学表达式为

+= (3.9)

可得Smith补偿器方程为

= (3.10)

于是，图3.2所示之Smith补偿系统又可表示为图3.3。

图3.3 Smith预估补偿系统一般型框图













图3.3经方等效变换，可转为如图3.4所示的方框图。

由图3.4显然可得等效Smith预估系统闭环传递函数为

 (3.11)

故闭环系统特征方程式为

=0 (3.12)











图3.4 Smith等效预估补偿系统框图

3.6 加热炉温度控制简介

工业过程控制系统中，温度控制是最为常见的控制情况，本文主要通过控制加热炉温度控制，来阐述优化控制思想，使用粒子群算法进行PID参数自整定优化。

3.7 加热炉系统特性分析

加热炉炉温温度控制是一个典型的带有纯滞后环节的控制对象，炉温滞后会带来对加热产品质量与产量的影响[10]。随着这些年数字计算机控制技术的发展，加热炉现如今都装备有数字计算机控制系统，以实现加热炉的燃烧控制[1]。加热炉良好的设计可以降低能耗，减少温室气体排放，提高产量与效率，现如今这项课题越来越被重视。

加热炉炉温的滞后性不是测量传感器系统的滞后，测量系统的滞后是因为测量取样过程传感器的特性造成的，对于这样产生的滞后可以用滞后补偿系统进行校正，然而加热炉温度的滞后是由于炉温变化速率低于燃料流量变化速率，这是加热炉的物理特性造成的，通常的滞后补偿器效果不好。

举例来说，将炉温由500℃加热到800℃，当炉温达到设定值后，由于其惯性作用，炉温会偏离设定值到升1000℃。这种情况是因为为炉温的控制是通过控制燃烧流量，而燃烧流量是具有滞后的，当控制系统得到期望与传感器炉温的偏差信号后，流量控制系统能够迅速做出反应，改变流燃烧量的大小，但由于炉温的滞后，尽管燃料流量产生了变化，但炉温并不会立即做出变化。这样，控制系统认为炉温偏差信号仍没有得到校正而进一步对燃料流量控制，最终导致了加热炉燃烧系统较大的输出超调量。这种超调量会严重浪费能源，增加企业成本，而且会降低产品产量。

因此，总结来说造成炉温具有较大超调量的原因，是因为炉温有了偏差后，控制炉温的燃料流量变化快，而实际炉温变化慢。在温度滞后的这段时间，温度偏差没有改变，因而控制系统输出的值仍按原来的偏差继续改变燃料的流量，由于。当炉温有反应时，燃料流量已超过了所需的设定值，从而引起燃料流量的浪费，造成炉温大幅度波动。

3.8 加热炉的传递函数

加热炉对象传递函数多为带有滞后环节的一阶系统，目前钢铁厂车间加热炉的传递函数和温度传感器及变送器的传递函数模型分别为：

(3.13)

(3.14)

第 4 章 系统仿真研究

4.1 工程上的参数整定

对于本文选中的加热炉模型。这里采用临界稳定法对系统进行初始整定，步骤如下：

1. 置调节器的积分时间为最大值，微分时间为0，比例带取适当大小的值，使系统进入运行状态。

2. 待系统运行稳定后，逐渐减小比例带，直到出现等幅振荡为止，即所谓的临界振荡过程。记录下此时的比例带值，并计算两个波峰的间距，记做。

3. 利用K和再按4.1表所给的计算公式，求出调节器各参数K，，的数值。

**表4.1 临界比例度法计算公式**

|  |  |  |  |
| --- | --- | --- | --- |
| *参数调节规律* | *比例带* | *积分时间* | *微分时间* |
| **P** | **2** | **\_\_** | **\_\_** |
| **PI** | **2．2** | **0．85** | **\_\_** |
| **PID** | **1．6** | **0．5** | **0．125** |

按如下结构对工程法优化的PID用MATLAB进行仿真：

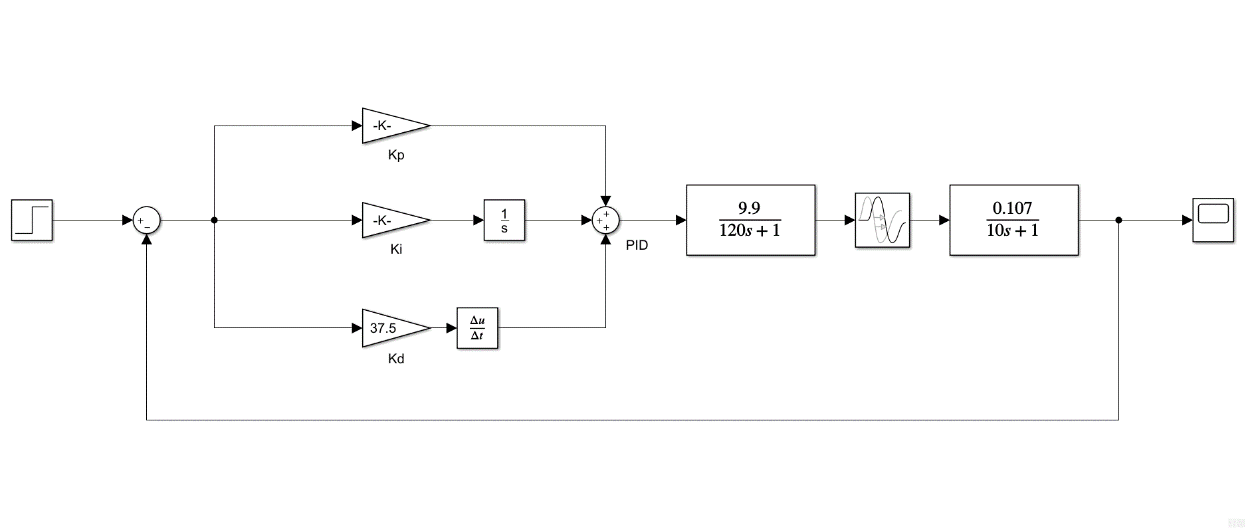


图4.1 加热炉温度的PID控制框图

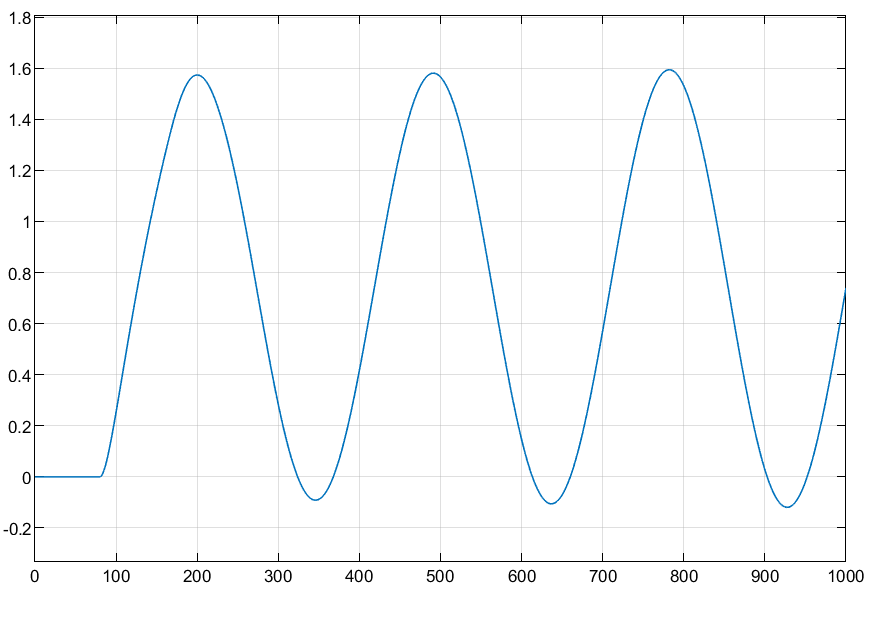


图4.2 单位阶跃响应的等幅振荡曲线

得到初始结果为：

=500-200=300

K=2.7

根据上表，可以计算用PID调节器调节时的各参数为：

=0.5926

=150，=0.0067

=37.5，=37.5

用工程整定后的仿真图形如下所示.取=0.5926，=0.0067，=37.5。

其仿真图形为图4.3所示。

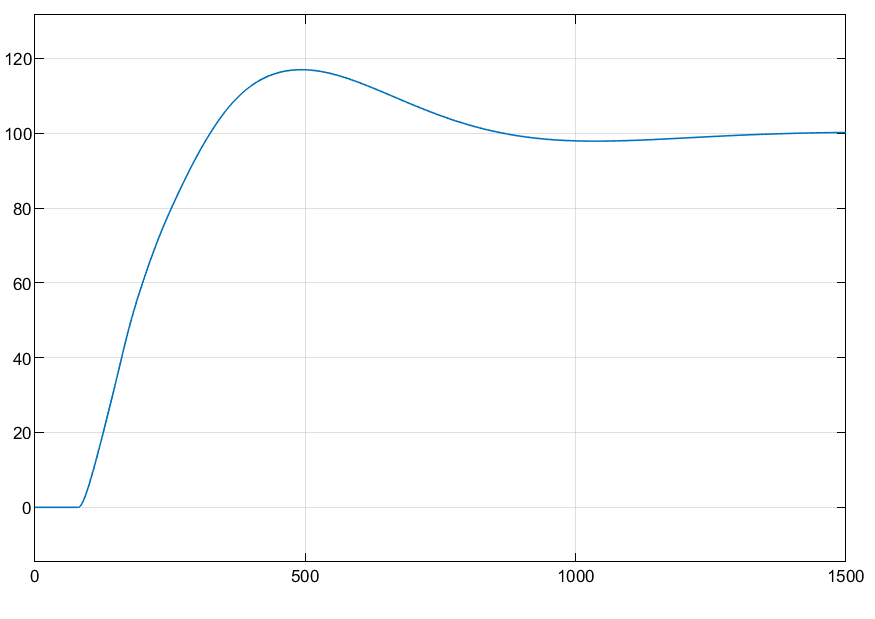


图4.3 用临界比例法整定后的曲线

4.2 粒子群算法参数整定

在普通PID控制模型的基础上，加上粒子群优化的目标函数，即3.4节介绍的带时间权值的目标损失函数，并且设置P、I、D三个参数为Kp、Ki、Kd三个位置变量。

设置粒子群优化的迭代次数为10次，粒子群规模为100，惯性因子为0.6，并且设置PID优化的范围，粒子群优化PID算法框图具体如图4.4所示。

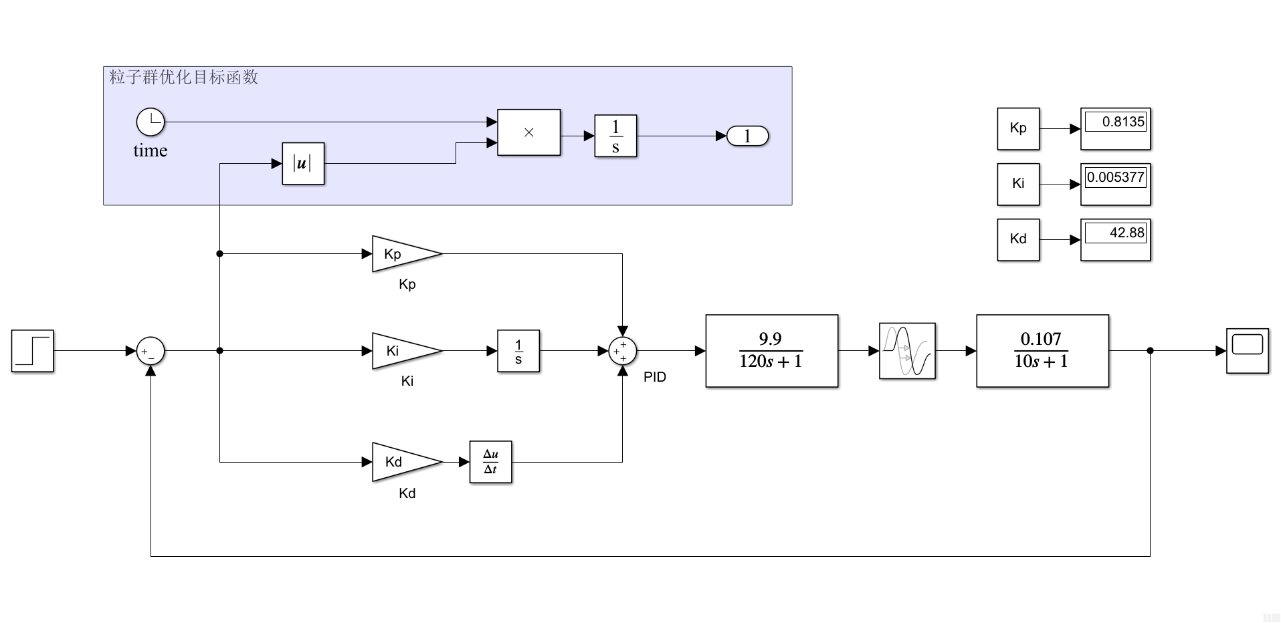


图4.3 用粒子群整定后的曲线

运行以上搭建的仿真模型，经过多次迭代优化，得出Kp、Ki和Kd变量的优化曲线，以及相应的随着迭代次数增加，个体适应值曲线。由图4.4可知，随着迭代次数的增加，粒子群算法动态改变PID参数，使得目标函数的适应值逐渐减小。



图4.4 粒子群优化PID迭代过程曲线

粒子群算法，经过10次迭代优化得到的最优PID参数为=1.2549，=0.0067，=41.4678，其优化后的系统输出相应曲线如图4.5所示。

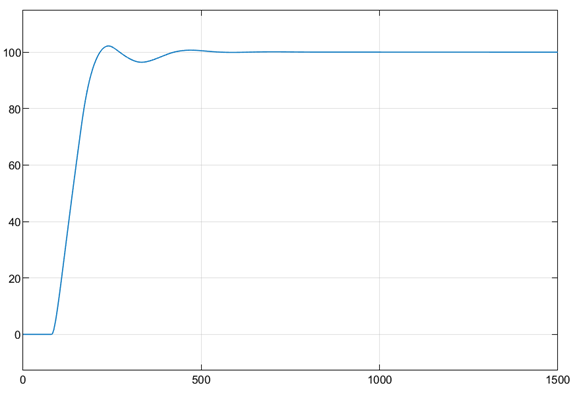


图4.5 粒子群优化PID输出相应曲线

4.3 结果比较

将以上两个模型进行汇总，以便用来比较临界比例法整定法和粒子群优化整定方法的各项性能，其Simulink仿真结果如图4.6所示。由图可知，粒子群优化的PID明显优于临界比例法整定的结果。前者阶跃响应曲线调节时间长、超调量大，容易造成过热影响产品质量，这对企业来说非常不利；而后者超调量非常小，系统调节时间很短，提高了系统控制性能，可以大大节省企业生产成本，提高效率和产品质量。

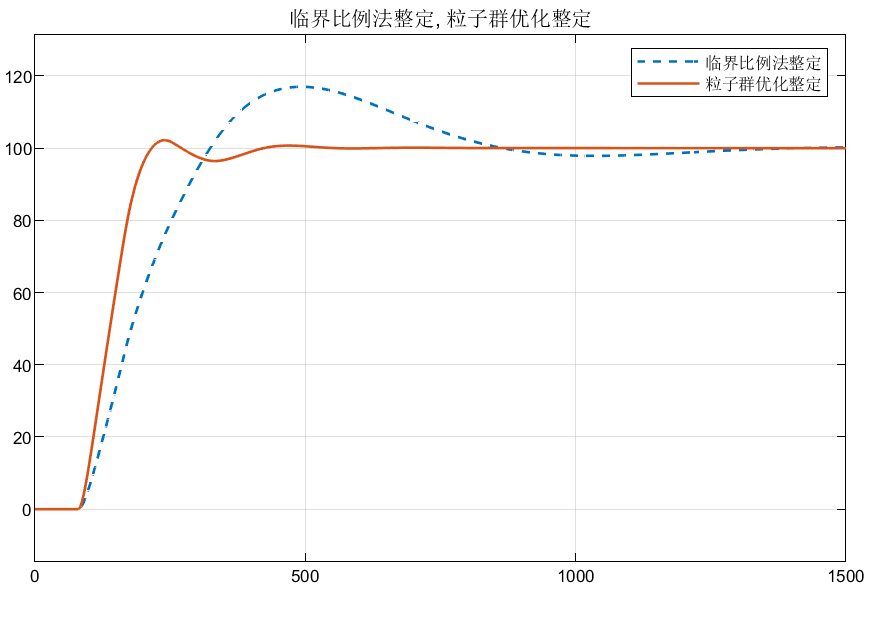


图4.6 两种不同方法的仿真曲线

总结

PID控制器被广泛应用在工业过程控制领域，其简单易懂、控制效果优良、无需知道受控对象数学模型，但随着现代工业的飞速发展对机器的精度的高要求，传统PID控制非线性、带滞后环节、高阶的甚至是时变的系统对象是非常困难的，常规PID很难跟踪理想的目标，效果欠佳。此外，在实际生产的实践中，常规PID控制器的整定过程往往是繁杂的、耗时耗力的，而且不好的PID参数容易出现，控制结果性能欠佳的情况，而本文提出的基于粒子群优化算法来优化传统PID控制参数对于整定PID的优越性是显而易见的。

本文以二阶带纯滞后时间对象的加热炉温度控制为例，详细介绍了粒子群算法的由来、算法原理流程、算法特点，重点使用了带惯性权值的粒子群算法来提高算法全局搜索能力。针对加热炉这种大滞后对象，采用误差绝对值积分型的目标函数作为适应度函数进行优化，粒子群算法经过多次的迭代，每次迭代都会更新仿真参数以减小目标函数的值，最终达到最优化。第四章仿真实验表明，经过PSO优化后的PID参数相比于领过工程经验整定后的参数具有更小的超调量、更少的调节时间和更好的稳定性。

致 谢

本次毕业设计论文全程都是在

参考文献

[1] 贠海坤,杨学友,尹宏瑛.智能控制工程在冶金生产机械中的应用[J].冶金管理,2019(15):114+116.

[2] 蔡自兴.人工智能在冶金自动化中的应用[J].冶金自动化,2015,39(01):1-8.

[3] Nieves Pavón-Pulido,Juan Antonio López-Riquelme,Jorge J. Feliú-Batlle. IoT Architecture for Smart Control of an Exoskeleton Robot in Rehabilitation by Using a Natural User Interface Based on Gestures[J]. Journal of Medical Systems,2020,44(9).

[4] Hee-Jin Kim,Guen-Han Dong,Dong-Ho Kim,Gi-Won Jang,Sung-Hyun Han. A Study on Track Record and Trajectory Control of Articulated Robot Based on Monitoring Simulator for Smart Factory[J]. Journal of The Korean Society of Industry Convergence,2020,23(2).

[5] Nieves Pavón-Pulido,Juan Antonio López-Riquelme,Jorge J. Feliú-Batlle. IoT Architecture for Smart Control of an Exoskeleton Robot in Rehabilitation by Using a Natural User Interface Based on Gestures[J]. Journal of Medical Systems,2020,44(9).

[6] 杜云超.PID过程控制及其参数整定策略浅析[J].化学工程与装备,2010(09):142-144.

[7] 冯琳欢. 输煤系统与磨煤机的控制与优化[D].新疆大学,2018.

[8] 潘峰,常嫦,马兵胜.时变大时滞系统的一种智能控制策略[J].电气技术,2009(07):19-22.

[9] Li Ruonan,Zhao Ping. Practical stability of time‐varying positive systems with time delay[J]. IET Control Theory & Applications,2021,15(8).

[10] 吕科东. 基于RBF神经网络PID控制在加热炉温度控制系统中的应用[A]. 天津市电子工业协会.天津市电子工业协会2019年年会论文集[C].天津市电子工业协会:天津市电子工业协会,2019:6.

附录 A（程序清单）

基本粒子群优化算法Matlab源程序

clear

clc

%% 参数设置

w = 0.6; % 惯性因子

c1 = 2; % 加速常数

c2 = 2; % 加速常数

Dim = 3; % 维数

SwarmSize = 100; % 粒子群规模

ObjFun = @PSO\_PID; % 待优化函数

MaxIter = 10; % 最大迭代次数

MinFit = 0.1; % 最小适应值

Vmax = 1;

Vmin = -1;

Ub = [1.5 0.01 100];

Lb = [0 0 0];

%% 粒子群初始化

Range = ones(SwarmSize,1)\*(Ub-Lb);

Swarm = rand(SwarmSize,Dim).\*Range + ones(SwarmSize,1)\*Lb % 初始化粒子群

VStep = rand(SwarmSize,Dim)\*(Vmax-Vmin) + Vmin % 初始化速度

fSwarm = zeros(SwarmSize,1);

for i=1:SwarmSize

fSwarm(i,:) = feval(ObjFun,Swarm(i,:)); % 粒子群的适应值

end

%% 个体极值和群体极值

[bestf bestindex]=min(fSwarm);

zbest=Swarm(bestindex,:); % 全局最佳

gbest=Swarm; % 个体最佳

fgbest=fSwarm; % 个体最佳适应值

fzbest=bestf; % 全局最佳适应值

%% 迭代寻优

iter = 0;

y\_fitness = zeros(1,MaxIter); % 预先产生4个空矩阵

K\_p = zeros(1,MaxIter);

K\_i = zeros(1,MaxIter);

K\_d = zeros(1,MaxIter);

while( (iter < MaxIter) && (fzbest > MinFit) )

for j=1:SwarmSize

% 速度更新

VStep(j,:) = w\*VStep(j,:) + c1\*rand\*(gbest(j,:) - Swarm(j,:)) + c2\*rand\*(zbest - Swarm(j,:));

if VStep(j,:)>Vmax, VStep(j,:)=Vmax; end

if VStep(j,:)<Vmin, VStep(j,:)=Vmin; end

% 位置更新

Swarm(j,:)=Swarm(j,:)+VStep(j,:);

for k=1:Dim

if Swarm(j,k)>Ub(k), Swarm(j,k)=Ub(k); end

if Swarm(j,k)<Lb(k), Swarm(j,k)=Lb(k); end

end

% 适应值

fSwarm(j,:) = feval(ObjFun,Swarm(j,:));

% 个体最优更新

if fSwarm(j) < fgbest(j)

gbest(j,:) = Swarm(j,:);

fgbest(j) = fSwarm(j);

end

% 群体最优更新

if fSwarm(j) < fzbest

zbest = Swarm(j,:);

fzbest = fSwarm(j);

end

end

iter = iter+1; % 迭代次数更新

y\_fitness(1,iter) = fzbest; % 为绘图做准备

K\_p(1,iter) = zbest(1);

K\_i(1,iter) = zbest(2);

K\_d(1,iter) = zbest(3);

end

%% 绘图输出

figure(1) %

plot(y\_fitness,'LineWidth',2)

title('最优个体适应值','fontsize',18);

xlabel('迭代次数','fontsize',18);ylabel('适应值','fontsize',18);

set(gca,'Fontsize',18);

figure(2)

subplot(3,1,1)

plot(K\_p,'LineWidth',3)

legend('Kp');

set(gca,'Fontsize',10);

subplot(3,1,2)

plot(K\_i,'k','LineWidth',3)

legend('Ki');

set(gca,'Fontsize',10);

subplot(3,1,3)

plot(K\_d,'r','LineWidth',3)

legend('Kd');

set(gca,'Fontsize',10);

suptitle('Kp、Ki、Kd 优化曲线');

set(gca,'Fontsize',10);

xlabel('迭代次数','Fontsize',12);