

doi: 10.7690/bgzdh.2026.04.012

改进遗传算法在火电厂主汽温控制系统中的研究

吴延凯, 魏宏飞

(河南工业职业技术学院自动化工程学院, 河南 南阳 473000)

摘要: 针对火电厂主汽温回路大惯性、大延迟、非线性及时变性的复杂特性, 且 PID 不能自整定及整定难的问题, 提出了一种基于改进遗传算法 (genetic algorithm, GA) 的并行式二自由度 PID 控制方法。运用提出的改进遗传算法, 对并行式二自由度 PID 控制器的参数进行优化, 并用于主汽温控制系统。仿真结果表明: 与其他方法相比, 该方法控制效果更优, 具有响应快、调节时间短、超调量小及鲁棒性好等特性, 可满足火电厂主汽温控制系统现场控制要求。

关键词: 主汽温; PID; 遗传算法; 二自由度; 超调量; 鲁棒性

中图分类号: TP273 **文献标志码:** A

Research on Improved Genetic Algorithm in Main Steam Temperature Control System of Thermal Power Plant

Wu Yankai, Wei Hongfei

(School of Automation Engineering, He'nan Polytechnic Institute, Nanyang 473000, China)

Abstract: In view of the complex characteristics of the main steam temperature loop in thermal power plants, such as large inertia, large delay, nonlinearity and time-varying, and the difficulty of PID self-tuning, a parallel two-degree-of-freedom PID control method based on improved genetic algorithm (GA) is proposed. The parameters of the parallel two-degree-of-freedom PID controller are optimized by using the improved genetic algorithm, and it is used in the main steam temperature control system. The simulation results show that compared with other methods, the proposed method has better control effect, fast response, short adjustment time, small overshoot and good robustness, which can meet the field control requirements of main steam temperature control system in thermal power plant.

Keywords: main steam temperature; PID; genetic algorithm; two-degree-of-freedom; overshoot; robustness

0 引言

在工业生产过程中, 温度控制是一类常见的控制对象, 而温度控制的精准、稳定与否直接关乎产品的质量。在火电厂发电过程中, 整个火电机组能够进行安全、经济及高效运转十分关键, 而其中起到决定性的因素是要保证主蒸汽温度的稳定^[1]。目前, 火电厂生产作业中, 对主汽温的控制要求很高, 汽温值需维持在额定温度 $-5\sim 5^{\circ}\text{C}$ 之间, 且波动不能太大, 这样才能保证发电机组正常稳定输出, 带来更大经济效益, 并且主汽温控制系统的过热器管道不会长期处于高温或者低温环境内而受损, 才能更长久、更稳定地工作^[2]。

当前主汽温控制回路一般采用串级 PID 控制, 该方法虽取得了不错的成效, 但是针对复杂多变的主汽温控制回路, 还是有所欠缺, 故找到适宜的控制方法来调节主汽温极为关键的。笔者在分析实际问题后, 提出一种基于改进 GA 的并行式二自由度

PID 控制方法, 提高了主汽温回路的控制效果。

1 主汽温系统模型建立

在火电机组安全高效运行过程中, 锅炉主汽温系统控制效果的好坏至关重要。控制系统的主要目标是让蒸汽管道流出的汽温值维持在规定范围内。此外, 对于该系统来讲, 被控对象具有大迟滞、大惯性非线性及时变性的复杂特点, 且汽温系统还随着机组负荷的改变而变化, 因此维持汽温值的稳定对于发电工艺来说极为关键^[3]。过热蒸汽工艺流程如图 1 所示。

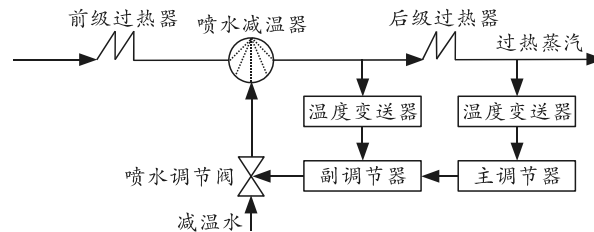


图 1 过热蒸汽工艺流程

收稿日期: 2024-12-10; 修回日期: 2025-01-05

基金项目: 河南省科技攻关项目(242102111191); 南阳市科技攻关项目(23KJGG073)

第一作者: 吴延凯(1992—), 男, 河南人, 硕士。

针对主汽温控制系统的复杂特性，可将主汽温控制回路的导前区和惰性区近似看作高阶对象环节。

$$W(s) = \frac{K}{(Ts + 1)^n} \quad (1)$$

式中： K 为放大系数； T 为对象时间常数。对于其模型参数的确定，可用高阶对象切线法来估算模型中的相关参数，如图 2 所示。

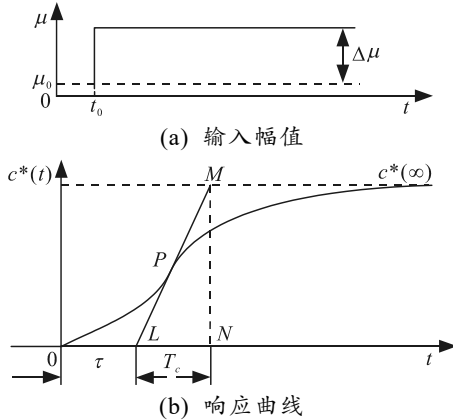


图 2 高阶对象切线法确定参数

系统模型中的相关参数求取具体为：

1) 作稳态值的渐近线 $c^*(\infty)$ ，则：

$$K = \frac{c^*(\infty) - c(0)}{\Delta\mu} \quad (2)$$

2) 由拐点 P 做切线交 $c^*(\infty)$ 渐近线于 M 点，交时间轴于 L 点，则： $\tau = OL$ ；线段 LM 在时间轴上的投影 $LN = T_c$ 。

3) n 和 T ：

$$\left. \begin{aligned} n &\approx 24 \times ((\tau/T_c + 0.12)/(2.93 - \tau/T_c)) \\ T &\approx (\tau + 0.5T_c)/(n - 0.35) \end{aligned} \right\} \quad (3)$$

2 并行式二自由度 PID 控制器

在工业控制过程中，PID 控制器被广泛应用。对于 PID 控制器来说，如果采用单自由度 PID 控制器，则 PID 参数只有一组可调节，其抑干扰特性、目标值跟踪特性不能同时达到最佳状态；追求抑干扰性能时，其目标值跟踪性能不能达到最优；反之，追求目标值跟踪性能时，其抑干扰性能就会降低^[4]。为解决此问题，需要找到一种可同时兼顾二者的方法，二自由度 PID 控制器应运而生。在实际控制过程中，其控制性能确实优于传统单自由度 PID^[5]。其优势表现为：在目标值跟踪和抑干扰性能上，二自由度 PID 可同时兼顾调整使其达到最佳状态，故采用二自由度 PID 控制器。笔者在一般的二自由度

PID 控制器的基础上，提出一种并行式二自由度 PID 控制器，原理如图 3 所示。

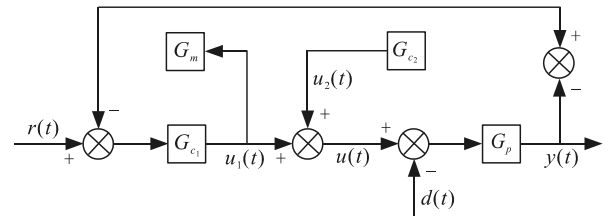


图 3 并行式二自由度 PID 控制结构原理

图 3 中： $r(t)$ 为系统输入； $d(t)$ 为扰动输入； $u_1(t)$ 为 PD 控制器的输出； $u_2(t)$ 为 PID 控制器的输出； $u(t)$ 为 2 个控制器输出的叠加； G_{c1} 为 PD 控制器； G_{c2} 为 PID 控制器； $y(t)$ 为系统输出^[6]。

对于 PD 控制器 G_{c1} 来讲，系统给定值的跟踪性能受其控制，其表达式为：

$$G_{c1}(s) = K_{c1}(1 + T_{d1}s) \quad (4)$$

式中： K_{c1} 为增益； T_{d1} 为微分常数。

对于 PID 控制器 G_{c2} 来讲，系统抗干扰能力受其控制，其表达式为：

$$G_{c2}(s) = K_{c2} \left(1 + \frac{1}{T_{i2}s} + T_{d2}s \right) \quad (5)$$

式中： K_{c2} 为增益； T_{d2} 为微分常数； T_{i2} 为积分常数。

则该二自由度的闭环输出表达式为：

$$y = \frac{G_p}{G_m} \left(\frac{1 + G_{c2}G_m}{1 + G_{c2}G_p} \right) \left[\frac{G_{c1}G_m}{1 + G_{c1}G_m} \right] r + \left[\frac{G_p}{1 + G_{c2}G_p} \right] d \quad (6)$$

在其参数整定过程中，通常取： $G_p = G_m$ ，则其可表示为：

$$y = \left[\frac{G_{c1}G_m}{1 + G_{c1}G_m} \right] r + \left[\frac{G_m}{1 + G_{c2}G_m} \right] d \quad (7)$$

3 改进 GA 融合算法的设计原理

3.1 PSO 算法原理

粒子群算法 (particle swarm optimization, PSO) 是受启于鸟群觅食行为，设计出的一种解决优化问题的算法^[7]。其具体的算法描述为：在 D 维的搜索空间中，有 n 个粒子构成一个种群 $x = (x_1, x_2, \dots, x_n)$ ；其中各粒子 i 的位置向量可表示为： $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ；速度向量可表示为： $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ；在 D 维的解空间中，粒子 i 在每次的迭代寻优过程中，会依据自身惯性、自身经验 $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ 和种群最优经验 $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ 来调整自身的迁移速度，进而调整自身的位置。在寻优过程中更新粒子的状态为：

$$v'_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}); \quad (8)$$

$$x'_{id} = x_{id} + v'_{id} \quad (9)$$

式中: $i=1, 2, \dots, n$; $d=1, 2, \dots, D$; c_1 和 c_2 为学习因子; r_1, r_2 为 $[0, 1]$ 之间的随机数; d 为迭代次数; ω 为惯性因子。

对于惯性因子 ω 来说, 作用是平衡局部、全局寻优, 若设定为固定值, 其过大过小都不可靠; 为此, 特提出了一种非线性递减的动态惯性权重^[8]。 ω 值随着优化迭代次数的增加而改变:

$$\omega = \omega_{\max} - \frac{\text{rand}(\omega_{\max} - \omega_{\min})}{\text{Maxiter}} \times \text{iter} \quad (10)$$

式中: iter 为当前的迭代次数; Maxiter 为最大的迭代次数。

3.2 改进 GA 融合算法原理

GA 和粒子群算法都是经典的群智能进化算法。对于粒子群算法来说, 容易早熟收敛而陷入局部最优; 对于遗传算法来说, 其全局搜索能力强, 但收敛速度较差。笔者在传统遗传算法基础上, 设计出一种改进 GA 的融合算法, 融合后的算法能弥补双方缺点, 增强优点, 使粒子群算法和遗传算法的性能均得到增强, 且与单一算法相比, 有着更好的全局优化能力^[9-11]。

在遗传算法中, 把每个可行解看作一条染色体, 表示为 $x^i = (x_1^i, x_2^i, \dots, x_n^i)$, 其中 i 为第 i 条染色体, n 为染色体的长度。改进 GA 算法的选择、交叉和变异算子的设计具体如下。

1) 选择算子。首先调整 PSO 的惯性权重, 更新出粒子的新位置和新速度, 接着将原个体与新个体进行适应度值比较, 若新个体优于原个体, 则保留新个体; 反之则保留原个体。然后将个体适应度值按降序方式排列, 然后据分类策略把种群个体均分为: 1 群体、2 群体和 3 群体。具体种群分类策略如图 4 所示。

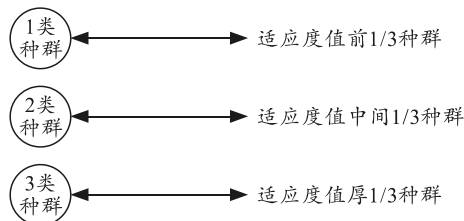


图 4 种群分类策略

2) 交叉算子。根据种群分类策略分为 1、2、3。以此为基础提出一种分类交叉策略来优化交叉操作, 以增加群体丰富度, 扩展搜索域, 提升全局搜

寻能力, 其操作步骤为:

Step1: 对于 1 种群, 采用单点交叉来进行交叉操作并保留父代个体;

Step2: 对于 2 种群, 采用单点交叉对本种群内部个体或与一类种群个体进行交叉操作;

Step3: 对于 3 种群, 采用两点交叉的方法, 交叉结束后, 按照适应度值降序原则将父代和新子代排序, 把适应度值高的个体保留下来。

3) 变异算子。变异操作是指在种群的某个体中, 基因发生突变以提高种群丰富度, 防止早熟收敛^[12]。笔者采用高斯变异来进行变异操作, 是一种对变异个体邻域局部搜索的算子, 该算子具有较强的局部搜索能力。其具体的操作方法是在个体的变异点位对应的基因点添加随机变化的干扰项, 并且该干扰项遵循高斯分布, 可以设变异的染色体为:

$$x^i = (x_1^i, x_2^i, x_3^i, \dots, x_n^i), \text{ 则变异后得到的新个体如下: } x'_i = x_i [1 + k * N(0, 1)] \quad (11)$$

式中: k 为 $(0, 1)$ 之间的随机数; $N(0, 1)$ 为均值是 0, 方差为 1 的高斯分布。

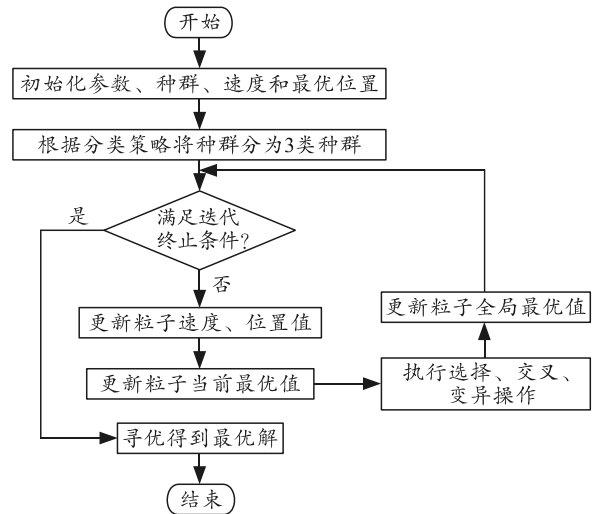


图 5 改进遗传算法流程

4 基于 Matlab 的仿真实例分析

4.1 改进 GA 在主汽温系统中的仿真实例

为验证文中提出方法的可行性, 将该方法运用到某火电厂主汽温控制回路中, 该对象传函还有大延迟环节, 为 e^{-90s} 。通过前面的建模及辨识, 其导前区及惰性区的传函分别如下:

$$W_1(s) = \frac{1.657}{(20s + 1)^2}; \quad (12)$$

$$W_2(s) = \frac{1.202}{(27.1s + 1)^7} \quad (13)$$

接下来要对改进 GA 执行参数寻优前的准备工作。将该改进二自由度 PID 的 5 个参数 K_{c1} 、 T_{d1} 、 $K_{c2}T_{i2}$ 以及 T_{d2} 进行实数编码，且各参数须满足： $K_{c1} \in (0, 1)$ 、 $T_{d1} \in (0, 1500)$ 、 $K_{c2} \in (0, 1)$ 、 $T_{i2} \in (0, 3000)$ 、 $T_{d2} \in (0, 1200)$ ，将这 5 个参数顺序进行连接组成单体。经测试实验，笔者所述问题一般取种群规模 $n=100$ 可达到理想优化效果。交叉概率取 0.85。为完成超调量、调节时间等时域指标的共同优化，用绝对误差积分准则来作为目标函数：

$$F = \int_0^{\infty} |e(t)| dt \quad (14)$$

选取适应度函数时，采用当前种群内最高的目标函数值 (IAE 值) 减去当前个体的目标函数值 (IAE 值) 以此当作适应度函数：

$$f = Obj_{\max} - Obj_i \quad (15)$$

4.2 实验验证分析

通过 Matlab 仿真，得到 5 种方法优化的主汽温系统单位阶跃响应曲线，如图 6 所示。

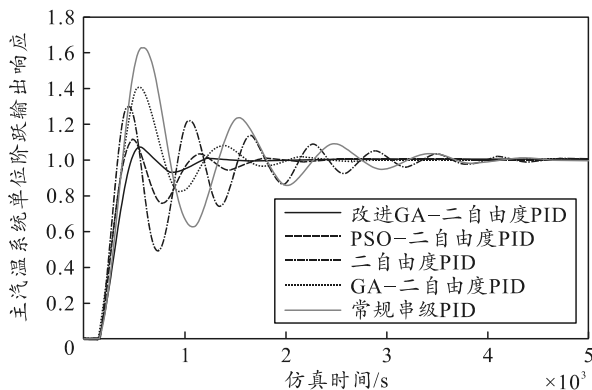


图 6 正常负荷时主汽温单位阶跃响应

通过仿真，得到改进 GA 的群体平均 IAE 随优化代数的迭代历程，如图 7 所示。

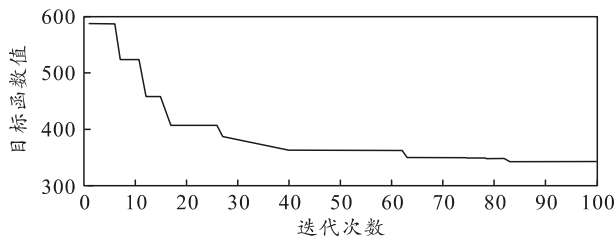


图 7 种群平均 IAE 值的优化过程

利用 IAE (绝对误差积分) 和 ISE (平方误差积分) 的计算公式，以及 Matlab 运算，可得到表 1 所示的 5 种方法的 4 个性能指标值。

$$IAE = \int_0^{\infty} |e(t)| dt \quad (16)$$

$$ISE = \int_0^{\infty} e^2(t) dt \quad (17)$$

表 1 性能指标值对比结果

| 调节方法 | $\sigma/\%$ | IAE | ISE | t_s/s |
|------------|-------------|-----------|-----------|-------------|
| 常规串级 | 62.100 0 | 755.657 3 | 396.640 4 | 2 628.000 0 |
| 并行式二自由度 | 30.300 0 | 626.261 4 | 312.954 3 | 2 359.000 0 |
| GA-二自由度 | 40.500 0 | 463.126 0 | 286.950 3 | 1 492.000 0 |
| PSO-二自由度 | 11.200 0 | 386.491 6 | 266.941 7 | 992.300 0 |
| 改进 GA-二自由度 | 6.800 0 | 343.263 4 | 261.881 0 | 630.200 0 |

通过表 1 及图 6 可知：常规串级 PID 控制效果最差，其超调量及调节时间最大，且 IAE 及 ISE 的值也最大，与其他 4 种控制方法的差距很明显。笔者提出的并行式二自由度 PID 的控制效果和串级 PID 对比，也具有很大优势；笔者提出的改进 GA-二自由度 PID 控制效果是最优的，其超调量小，调节时间短，IAE 及 ISE 的值也很小，性能指标值最优。

4.3 鲁棒性测试

由于火电厂锅炉主汽温控制系统，工况复杂多变，不确定因素多，存在模型失配的问题。为进行鲁棒性测试，仍以上述方法得到的参数为基础，进行负荷增加和负荷降低的实验。

当系统负荷增加时，主汽温导前区和惰性区传函为：

$$W_1(s) = \frac{1.3256}{(16s+1)^2}; \quad (18)$$

$$W_2(s) = \frac{1.442}{(21.68s+1)^7} \quad (19)$$

运用上 5 种控制方法，对参数扰动后的主汽温对象进行仿真实验，得到的负荷增加时系统单位阶跃响应曲线如图 8 所示。

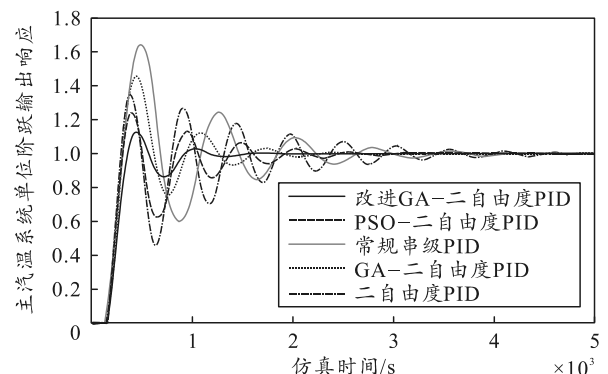


图 8 负荷增加时主汽温单位阶跃响应曲线

当系统负荷降低时，主汽温特性的导前区和惰性区传函如下：

$$W_1(s) = \frac{1.988}{(24s+1)^2}; \quad (20)$$