

文章编号: 1001-0920(2011)11-1731-04

基于改进混合蛙跳算法的电渣重熔过程 多变量PID控制器设计

王介生^{1,2}, 高宪文¹

(1. 东北大学 信息科学与工程学院, 沈阳 110819; 2. 辽宁科技大学 电子与信息工程学院, 辽宁 鞍山 114044)

摘要: 根据电渣重熔过程的工艺特点和数学模型, 提出了基于改进混合蛙跳算法(ISFLA)的多变量参数自整定PID控制策略. 提出一种新的蛙跳规则, 用以增强SFLA的局部搜索能力. 该规则主要通过模拟青蛙的感知和运动的不确定性来动态随机地调整青蛙的局部搜索空间和步长, 以防止SFLA算法过早收敛, 提高算法的搜索效率. 仿真结果和工业应用实验均表明了所提出控制方法的可行性和有效性.

关键词: 电渣重熔过程; 多变量系统; PID控制器; 混合蛙跳算法

中图分类号: TP273.2

文献标识码: A

Design of multivariable PID controller of electroslag remelting process based on improved shuffled frog leaping algorithm

WANG Jie-sheng^{1,2}, GAO Xian-wen¹

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 2. School of Electronic and Information Engineering, Liaoning University Science and Technology, Anshan 114044, China. Correspondent: WANG Jie-sheng, E-mail: wang_jiesheng@126.com)

Abstract: Based on the technique features and mathematical model of electroslag remelting(ESR) process, a multivariable self-tuning PID controller tuned optimally by an improved shuffled frog leaping algorithm(ISFLA) is proposed to control the two-input-two-output(TITO) ESR process. A new frog leaping rule is proposed to enhance the SFLA's local search capabilities, which adjusts the frog local search space and the step of each frog's jump dynamically and randomly by emulating frog's perception and action uncertainties in order to prevent premature convergence and improve the search efficiency of SFLA. The simulation results and industrial application tests show the feasibility and effectiveness of the proposed control method.

Key words: electroslag remelting process; multivariable system; PID controller; shuffled frog leaping algorithm

1 引言

电渣重熔(ESR)是在初炼钢的基础上进一步提纯钢质, 得到均匀致密、纵向结晶的优质钢的冶炼方式. 电渣炉早期的控制主要采用电压摆动、恒电流、恒电压和递减功率控制等方法. 文献[1]分析研究了电渣重熔模型, 对重熔电极熔速和位置进行协同控制, 应用改进的遗传算法对协同控制器参数寻优. 文献[2]针对ESR冶炼工艺对电极控制系统的要求, 提出了一种模糊自适应PID控制方法. 这些控制策略都只对熔炼电流和电压进行单独回路控制, 没有考虑熔炼电流与熔炼电压之间的耦合关系, 以及主回路中短

网阻抗、渣阻和机械特性对电渣重熔过程控制性能的影响.

PID控制器已被广泛应用于工业过程控制领域, 而其参数的整定与优化一直是一个重要的问题^[3]. 随着群集智能理论的发展, 出现了基于遗传算法^[4]和粒子群算法^[5]的先进PID控制器参数整定方法. 但是, 遗传算法需要进行复制、交叉和变异操作, 其性能对参数有较大的依赖性; 粒子群优化算法容易陷入局部最优, 产生早熟现象.

混合蛙跳算法(SFLA)是一种基于群体的亚启发式协同搜索群智能算法, 已在水资源网络优化^[6]、贴

收稿日期: 2010-08-18; 修回日期: 2010-11-14.

基金项目: 中国博士后科学基金面上项目(20110491510); 辽宁省教育厅创新团队基金项目(2008T091).

作者简介: 王介生(1977-), 男, 副教授, 博士后, 从事复杂工业过程建模的研究; 高宪文(1955-), 男, 教授, 博士生导师, 从事智能控制、复杂工业过程建模等研究.

装顺序优化^[7]等方面得到应用. 为此, 本文根据电渣重熔过程动态特性及其数学模型, 提出一种基于改进混合蛙跳算法的多变量参数自整定PID控制策略.

2 基于ISFLA的多变量PID控制器

2.1 多变量PID控制策略

PID控制是按照偏差的比例、微分和积分的线性组合进行控制的一种调节器, 可描述为

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + \frac{T_d de(t)}{dt} \right]. \quad (1)$$

其中: $K_i = K_p/T_i$, $K_d = K_p T_d$, $e(t)$ 为反馈偏差.

本文提出的电渣重熔过程多变量PID控制器的结构如图1所示. 根据被控对象的特点和现场工况, 针对输出2的PID控制器不能包含微分环节, 否则系统将发散, 因此系统由PID控制器(PID₁和PI₂)与被控对象组成. 鉴于PID控制器设计实际上是多维函数优化问题, 混合蛙跳算法采用实数编码, 对于电渣重熔过程多变量PID控制器参数可直接编码为

$$X = \{ \underbrace{K_{p1}, K_{i1}, K_{d1}}_{PID_1}, \underbrace{K_{p2}, K_{i2}}_{PI_2} \}. \quad (2)$$

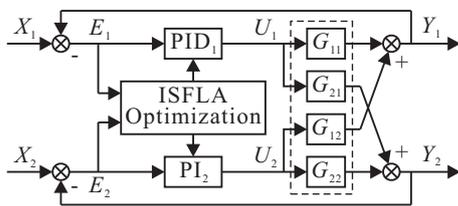


图1 参数自整定PID解耦控制系统结构

控制参数优化旨在使系统总体控制偏差趋于零、有较快响应速度和较小的超调量, 适应度函数采用IAE. 针对多变量系统, 定义如下^[8]:

$$F_{IAE} = \int_0^{T_f} \rho_i |e_i(t)| dt. \quad (3)$$

式中: ρ_i ($i=1, 2$) 反映了第 i 个输出的权重因子, T_f 为仿真结束时间. PID控制器参数由改进混合蛙跳算法进行全局优化, 其目标是

$$\min_X F_{IAE}(X). \quad (4)$$

2.2 混合蛙跳算法

SFLA是一种结合了确定性方法和随机性方法的进化计算方法. 其基本思想是^[6]: 随机生成 N 只蛙, 组成初始种群 $P = \{X_1, X_2, \dots, X_N\}$, S 维解空间中的第 i 只蛙表示为 $X_i = [x_{i1}, x_{i2}, \dots, x_{iS}]$. 生成初始群体之后, 将种群内蛙的个体按适应值降序排列, 记录蛙群中具有最优适应值的蛙为 X_g ; 然后将整个蛙群分成 m 个模因组, 每个模因组包含 n 只蛙, 满足关系 $N = m \times n$, 其中第 1 只蛙分入第 1 模因组, 第 2 只蛙分入第 2 模因组, 第 m 只蛙分入第 m 模因组, 第 $m+1$ 只蛙重新分入第 1 模因组, 依次类推, 设 M^k 为第 k 个模因组的蛙的集合, 其分配过程描述如下:

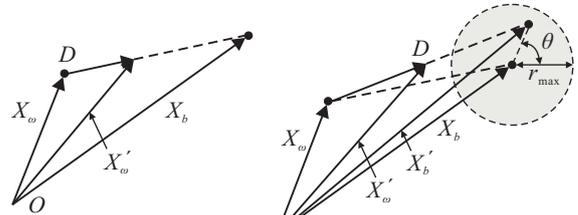
$$M^k = \{X_{k+m(l-1)} \in P | 1 \leq l \leq n\}, 1 \leq k \leq m. \quad (5)$$

每一个模因组中的最好适应值和最差适应值的蛙分别记为 X_b 和 X_w , 而种群中具有最好适应值的蛙表示为 X_g ; 然后对每个模因组进行局部搜索, 即对模因组中的 X_w 循环进行局部搜索操作. 根据最初蛙跳规则(如图2(a)所示), 其更新方式为

$$D = r \cdot (X_b - X_w); \quad (6)$$

$$X'_w = X_w + D, \|D\| \leq D_{\max}. \quad (7)$$

式中: r 表示 0 与 1 之间的随机数, D_{\max} 表示蛙所允许改变位置的最大值. 在经过更新后, 如果得到的蛙 X'_w 优于原来的蛙 X_w , 则取代原来模因组中的蛙; 如果没有改进, 则用 X_g 取代 X_b , 按式(6)和(7)执行局部搜索过程; 如果仍然没有改进, 则随机产生一个新蛙直接取代原来的 X_w . 重复上述局部搜索 L_{\max} 次, 当完成局部搜索后, 将所有模因组内的蛙重新混合并排序和划分模因组, 再进行局部搜索, 如此反复, 直到达到混合迭代次数 G_{\max} .



(a) 原蛙跳规则

(b) 改进蛙跳规则

图2 蛙跳规则示意图

2.3 改进蛙跳规则

在蛙群的自然模因进化中, 较差的蛙受较好蛙的影响而为了获得更多的食物跳向较好的蛙. 根据上述的初始蛙跳规则(如图2(a)所示), 最差蛙的可能的新的位置被限定在当前值与最好蛙位置的线段上, 这种规则限制了模因进化的搜索区域, 不仅降低了收敛速度, 而且易导致早熟收敛. 文献[8]提出了一种基于新蛙跳规则的混合蛙跳算法(MSFLA), 可表示为

$$D = r \cdot c \cdot (X_b - X_w) + W; \quad (8)$$

$$W = [r_1 w_{1,\max}, r_2 w_{2,\max}, \dots, r_S w_{S,\max}]^T; \quad (9)$$

$$X'_w = \begin{cases} X_w + D, \|D\| \leq D_{\max}; \\ X_w + \frac{D}{\sqrt{D^T D}} D_{\max}, \|D\| > D_{\max}. \end{cases} \quad (10)$$

式中: r 为 $[0, 1]$ 之间的随机数, c 为 $[1, 2]$ 之间的一个常数, r_i ($1 \leq i \leq S$) 为 $[-1, 1]$ 之间的随机数, $w_{i,\max}$ ($1 \leq i \leq S$) 为第 i 维搜索空间的最大感知和运动的不确定性, 此种蛙跳规则在一定程度上增大了算法搜索范围. 本文结合SFLA的特性, 提出一种新的如图2(b)所示的蛙跳规则, 表示为

$$X'_b = X_b + r_1 \cdot r_{\max} e^{j\theta}, \quad (11)$$

$$D = r_2 \cdot (X'_b - X_w). \quad (12)$$

式中: θ 为 $[0, 360^\circ]$ 之间的随机旋转角度, r_{\max} 为最大局部搜索半径, r_1 和 r_2 均为 $[0, 1]$ 之间的随机数. r_{\max} 决定了蛙在模因组中局部优值周边的最大感知; θ , r_1 和 r_2 决定了蛙跳跃的不确定性. 蛙的位置向量仍然采用式 (10) 进行更新.

2.4 算法流程

结合所提出的蛙跳规则, 参考文献 [7] 的算法流程, 基于 ISFLA 算法整定 PID 控制器的算法流程如下:

Step 1: ISFLA 算法参数初始化. 初始化蛙种群大小 N , 搜索空间维数 S , 模因组数 m , 则每个模因组包含 n 只蛙 ($N = m \times n$), 蛙所允许改变的位置最大值为 D_{\max} , 局部搜索次数为 L_{\max} , 全局混合迭代次数为 G_{\max} , 最大局部搜索半径为 r_{\max} .

Step 2: 随机生成 N 只蛙, 组成初始种群 $P = \{X_1(t), \dots, X_k(t) \dots, X_N(t)\}$, $k = 1, 2, \dots, N$, 迭代计数值 $t = 0$; 将每只蛙 $X_k(t)$ 作为 TITO 系统 PID 控制器参数, 然后对系统进行仿真, 根据式 (3) 计算适应度值 $F_k(t) = F(X_k(t))$; 对蛙群按适应度值进行递增方式排序, 并以 $U_k(t) = \{X_k(t), F_k(t)\}$ 形式存储, 记录蛙群最优蛙为 $X_g(t) = U_1(t)$.

Step 3: 模因组生成. 将 U 分成 m 个模因组 $M^1(t), \dots, M^j(t), \dots, M^m(t)$, $j = 1, 2, \dots, m$, 每个模因组包含 n 只蛙, 根据式 (5) 进行分配, 并记录模因组内最优蛙和最差蛙为 $X_b^j(t)$ 和 $X_w^j(t)$.

Step 4: 模因进化. 对模因组 $M^j(t)$ 中的最差蛙 $X_w^j(t)$ 分别按本文提出的新蛙跳规则进行局部搜索, 即根据式 (11) 和 (12) 决定蛙跳步长, 根据式 (1) 进行位置更新, 将其作为 TITO 系统 PID 控制器参数, 仿真并计算适应度值. 如果更新后的蛙优于原来的蛙, 则取代原来模因组中的蛙; 如果没有改进, 则用 $X_g(t)$ 取代 $X_b^j(t)$, 按式 (10)~(12) 执行局部搜索; 如果仍然没有改进, 则随机产生一个新蛙直接取代原来的 $X_w^j(t)$. 重复上述局部搜索 L_{\max} 次, 得到蛙已经被进化的模因组 $M^1(t)', M^2(t)', \dots, M^m(t)'$.

Step 5: 模因组混合. 将更新后的模因组 $M^1(t)', M^2(t)', \dots, M^m(t)'$ 内的蛙重新混合, 记 $U(t+1) = \{M^1(t)', M^2(t)', \dots, M^m(t)'\}$, 对 $U(t+1)$ 中的蛙按适应度值进行递增方式排序, 更新蛙群最优蛙为 $X_g(t+1) = U_1(t+1)$. 检测算法终止条件 $t = t+1$, 如 $t < G_{\max}$, 则转 Step 3; 否则, 输出最优蛙.

3 实验和结果分析

3.1 电渣重熔过程数学模型

在电渣重熔过程处于稳定工况的前提下, 分别给出二次电压 V_2 , 熔炼电流 I_2 的阶跃扰动, 得到励磁电压 ΔV 和熔速 v 的阶跃响应曲线, 在考虑变量量纲的

前提下, 通过曲线拟合方法建立电渣炉. 该工况下的传递函数矩阵为

$$\begin{bmatrix} \Delta V(S) \\ v(S) \end{bmatrix} = \begin{bmatrix} G_{11}(S) & G_{12}(S) \\ G_{21}(S) & G_{22}(S) \end{bmatrix} \begin{bmatrix} V_2(S) \\ I_2(S) \end{bmatrix} = \begin{bmatrix} \frac{1.38}{20s+1} & \frac{-0.32}{5s+1} - \frac{0.54}{s+0.9} \\ -0.17 & \frac{1.16}{30s+1} e^{-\tau s} \end{bmatrix} \begin{bmatrix} V_2(S) \\ I_2(S) \end{bmatrix}. \quad (13)$$

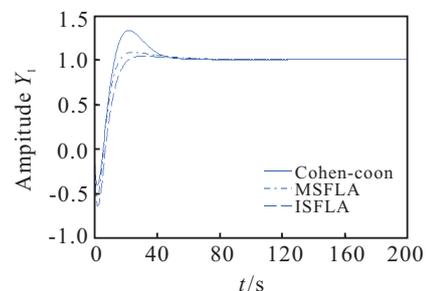
电流的周期性波动由滞后时间变量 τ 决定, 而 τ 由主回路中的阻抗变化决定, 滞后时间一般为 $3 \sim 6$ s, 本文取 $\tau = 3$.

3.2 算法仿真

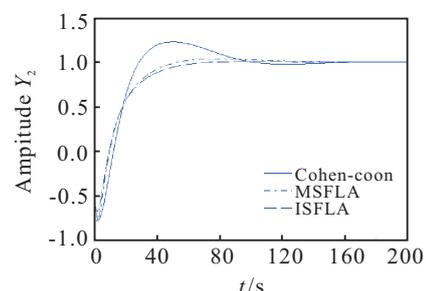
分别采用 Cohen-coon 工程整定方法、MSFLA 算法^[8]和本文提出的 ISFLA 算法整定图 1 所示的电渣重熔过程 TITO 系统的 PID 控制器参数, 比较阶跃响应下的控制效果. 在优化整定 PID 参数中, K_p 的搜索区间为 $[0, 3]$, K_i 和 K_d 的搜索区间为 $[0, 1]$.

MSFLA 和 ISFLA 算法选取相同的控制参数: 蛙种群大小 $N = 50$, 搜索空间维数 $S = 5$, 模因组数 $m = 5$, 每个模因组包含蛙数 $n = 10$, 蛙所允许改变位置最大值 $D_{\max} = 0.02$, 局部搜索次数 $L_{\max} = 5$, 全局混合迭代次数 $G_{\max} = 100$, ISFLA 算法中的最大局部搜索半径 $r_{\max} = 0.01$, MSFLA 算法中的运动不确定参数 $w_1 = w_2 = 0.01$, $w_3 = w_4 = w_5 = 0.001$.

图 3 为针对电渣重熔过程, 采用图 2 所示的控制策略, 使用 Cohen-coon, MSFLA 和 ISFLA 方法进行 PID 参数整定后闭环阶跃响应. 从图 3 的比较结果可以看出, MSFLA 和 ISFLA 优化算法均能在有限的进化代数内对电渣重熔被控对象得到满意的 PID 控制器参



(a) 输出1的响应曲线



(b) 输出2的响应曲线

图 3 不同 PID 控制器参数整定方法下的输出响应

数,但ISFLA具有更好的全局搜索能力.综合考虑系统的超调量、上升时间和调节时间这些性能指标,ISFLA算法比Cohen-coon方法和MSFLA算法具有更高的控制性能.

3.3 工业应用

所提出的基于ISFLA的多变量PID控制方法的工业应用现场环境为某特种钢厂抽锭式5T电渣炉.工况为 $\Phi 720$ 型结晶器,抽锭起始速度为 12 mm/min ,自耗电极 $\Phi 200$,冶炼电压设定值为 56 V ,冶炼电流设定值为 15000 A ,渣量 170 kg ,交换三次电极.以冶炼电流为例,图4是分别采用PID控制器和本文提出的多变量PID控制器的实时控制曲线.从图4可以看出,本文提出的智能控制策略使熔炼电流和熔炼电压的波动幅度明显减小,控制精度得到显著提高,熔铸被控制在工艺要求范围之内,有效地提高了铸锭质量.

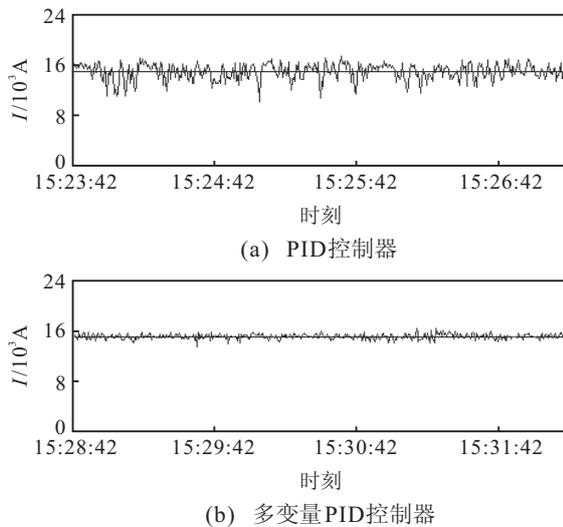


图4 电渣重熔实时控制曲线

4 结论

本文针对电渣重熔过程的特点,基于其数学模型,提出了基于改进混合蛙跳算法优化的TITO系统参数自整定PID控制策略.采用本文提出的ISFLA局部搜索空间的动态调整,可有效地防止算法早熟收敛,提高算法效率.仿真结果和工业应用实验结果表明,所提出的控制策略具有较好的动态和稳态性能,控制精度提高了 $1\% \sim 3\%$,有效地降低了铸造单位电耗.

参考文献(References)

- [1] 赵丽丽, 宋锦春, 刘喜海, 等. 基于遗传算法的电渣炉重熔过程智能控制研究[J]. 机械与电子, 2008, 17(5): 16-19.
(Zhao L L, Song J C, Liu X H, et al. Intelligent control study based on genetic algorithm for electroslag remelting process[J]. Machinery & Electronics, 2008, 17(5): 16-19.)
- [2] 任伟, 郑险峰, 姜立新, 等. 电渣炉电极调节系统的模糊自适应PID控制[J]. 冶金自动化, 2006, 9(1): 15-18.
(Ren W, Zheng X F, Jiang L X, et al. Fuzzy adaptive PID control for electrode regulating system of electroslag furnace[J]. Metallurgical Industry Automation, 2006, 9(1): 15-18.)
- [3] Chanchal D, Rajani K M. An improved auto-tuning scheme for PID controllers[J]. ISA Transactions, 2009, 48(4): 396-409.
- [4] Cheng C H, Cheng P G, Xie M J. Current sharing of paralleled DC-DC converters using GA-based PID controllers[J]. Expert Systems with Applications, 2010, 37(1): 733-740.
- [5] Chan W D, Shih S P. PID controller design of nonlinear systems using an improved particle swarm optimization approach[J]. Communications in Nonlinear Science and Numerical Simulation, 2010, 15(11): 3632-3639.
- [6] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. J of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [7] 朱光宇. 模因内三角概率选择混合蛙跳算法[J]. 计算机集成制造系统, 2009, 15(10): 1979-1985.
(Zhu G Y. Meme triangular probability distribution shuffled frog-leaping algorithm[J]. Computer Integrated Manufacturing Systems, 2009, 15(10): 1979-1985.)
- [8] Huynh T H. A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers[C]. IEEE Int Conf on Industrial Technology. Perth: IEEE Press, 2008: 1-6.

(上接第1730页)

[9] Zhou D Y, Bousquet O, Lal T N, et al. Learning with local and global consistency[C]. Proc of Advances in Neural Information Processing Systems. Massachusetts: MIT Press, 2003: 321-328.

[10] Wang F, Zhang C S. Label propagation through linear neighborhoods[J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20(1): 55-67.