

基于改进粒子群算法的PID参数优化方法研究

孔明^{1†}, 张琳², 张龙龙¹

(1. 山东新华医疗器械股份有限公司, 山东 淄博 255000; 2. 山东新华联合骨科器材股份有限公司, 山东 淄博 255000)

摘要:为解决粒子群优化(PSO)算法探索和利用不平衡导致寻优能力下降问题,提出了一种改进的PSO(IPSO)算法。首先,引入莱维飞行建立种群初始搜索模型提升算法探索能力;其次,建立种群探索行为模型、种群开发行为模型和探索与开发的平衡模型,自适应地平衡寻优过程中算法的探索和利用性能;最后,设计基于IPSO算法的直流电机PID控制器参数优化流程。实验结果表明,相较于对比算法,IPSO算法在CEC 2017函数问题上是最优算法,基于IPSO算法的PID控制器具有更优的控制性能,提出的方法为工程中PID控制器的参数优化提供了参考。

关键词: PID; PSO算法; 探索和利用; 莱维飞行; 参数优化

中图分类号: TP391

文献标识码: A

Study on PID Parameter Optimization Method Based on Improved Particle Swarm Optimization Algorithm

KONG Ming^{1†}, ZHANG Lin², ZHANG Longlong¹

(1. Shinva Medical Instrument Co., Ltd., Zibo, Shandong 255000, China;

2. Shandong Xinhua United Orthopedic Equipment Co., Ltd., Zibo, Shandong 255000, China)

Abstract: In order to solve the problem of decreasing the optimisation search ability due to the imbalance between exploration and exploitation of particle swarm optimisation (PSO) algorithms, an improved PSO (IPSO) algorithm is proposed. Firstly, Lévy flight is introduced to establish a population initial search model to enhance the algorithm's exploration ability; secondly, a population exploration behaviour model, a population exploitation behaviour model and an exploration-exploitation equilibrium model are established to adaptively balance the algorithm's exploration and exploitation performances in the optimization process; and lastly, a parameter optimization process of the PID controller of a DC motor based on the IPSO algorithm is designed. The experimental results show that the IPSO algorithm is the best algorithm in the CEC 2017 function problem compared to the comparison algorithm, and the PID controller based on the IPSO algorithm has a better control performance, and the proposed method provides a reference for the parameter optimisation of PID controllers in engineering.

Key words: PID; PSO algorithm; exploration and exploitation; Lévy flight; parameter optimization

比例积分微分(Proportional Integral Differential, PID)控制器是一种应用广泛的机器人控制器,具有结构简洁、鲁棒性强和布置方便等优势^[1-4]。PID控制器包括三个重要参数,分别是比例参数、积分参数和微分参数,参数与受控系统的

匹配程度直接决定了PID控制器的控制效果。目前,PID参数的整定方法可主要分为两类,分别是经验整定方法和基于智能算法的整定方法^[5]。相较于经验整定方法,基于智能算法的整定方法具有时间成本低和调节精度高的优势。

收稿日期:2024-10-29

作者简介:孔明(1990—),男,山东淄博人,硕士研究生,工程师,研究方向:元启发式算法设计与开发,医用清洗消毒器控制系统的规划、运行与维护。

†通信联系人,E-mail:1031358428@qq.com

基于群体的元启发式算法是一类经典的智能算法,因其简洁、易实现且寻优性能优异被广泛地用于PID参数优化任务^[6-9],改进算法性能有益于提升PID参数整定效果,进而提升PID控制器的控制效果。为了获取更优质的PID参数,该领域的研究人员围绕算法改进方法展开了大量研究工作。吴沁等^[10]提出了一种二阶振荡混沌映射粒子群算法,优化后的BP-PID控制器提高了滚珠丝杠进给伺服系统的跟踪精度和控制稳定性。张稚荷等^[11]提出了一种改进的麻雀搜索算法,优化了外骨骼关节PID控制器,减少了外骨骼机器人关节跟踪误差。舒奕彬等^[12]提出了一种改进的海马优化算法并设计了配套的PID优化方法,提高了系统控制精度。ZHANG等^[13]提出了一种基于CRICLE混沌映射初始化的改进鲸鱼优化模糊PID控制策略,降低了超调量并显著改善了调节时间。目前,算法改进的方向主要集中在初始种群和自适应参数,种群自身的搜索行为和每个阶段的探索和利用的平衡关注较少,然而这两方面的改进对算法性能的提升是不可忽略的^[14]。与所有非凸优化问题一样,调整PID算法以实现精确稳定的闭环控制成为一个NP-Hard问题^[15]。如何通过提升算法性能提高PID控制器的控制效果,仍然是一个值得研究的课题。

提出了一种基于改进的PSO(Improved Particle Swarm Optimization, IPSO)算法的PID参数整定方法。完善算法寻优过程,建立粒子搜索行为模型,平衡了各个阶段的探索和开发,有效地提高了算法的探索能力和利用能力。将IPSO算法应用在直流电机的PID控制器参数优化任务,评估并对比了不同算法优化后的PID控制器效果。

1 PID控制器

PID控制是一种应用广泛的控制策略,由比例环节、积分环节和微分环节相互耦合产生控制效果。PID控制器的控制量 u 由式(1)计算。

$$u(t_{pid}) = K_p e(t_{pid}) + K_i \int_0^{t_{pid}} e(t_{pid}) dt + K_d \frac{d(e(t_{pid}))}{dt_{pid}} \quad (1)$$

式中, K_p 是比例环节系数, K_i 是积分环节系数, K_d 是微分环节系数, $e()$ 是当前时刻的误差。

2 粒子群优化算法

标准粒子群优化算法模拟了鸟群觅食行为,是

一种经典的基于种群的元启发式算法,擅长解决连续域寻优问题,种群中粒子更新公式为式(2)。

$$\begin{aligned} \mathbf{V}_{i,j}^{t+1} &= \boldsymbol{\Omega}_{i,j}^t \mathbf{V}_{i,j}^t + C_1 (\mathbf{p}_{best} - \mathbf{X}_{i,j}^t) + \\ &C_2 (\mathbf{g}_{best} - \mathbf{X}_{i,j}^t) \\ \mathbf{X}_{i,j}^{t+1} &= \mathbf{X}_{i,j}^t + \mathbf{V}_{i,j}^{t+1} \end{aligned} \quad (2)$$

式中, t 是迭代次数, \mathbf{X}^t 是当前种群位置矩阵, \mathbf{V}^t 是当前种群速度矩阵, $\boldsymbol{\Omega}^t$ 是粒子速度的权重矩阵, \mathbf{X}^{t+1} 是 $t+1$ 次迭代的种群位置, \mathbf{V}^{t+1} 是 $t+1$ 次迭代的种群速度矩阵, $i \in [1, N]$, $j \in [1, D]$, D 是目标函数的维数, N 是种群大小, C_1 和 C_2 分别是个体认知常数和社会认知常数。

3 改进的粒子群算法

标准粒子群算法只考虑了全局最优粒子和个体历史最优粒子对种群的影响,种群的搜索行为对算法性能的影响被忽略了,改进的种群搜索行为叙述如下。

3.1 种群觅食行为模型

3.1.1 种群初始搜索阶段

鸟群觅食的初始阶段由目标位置的不确定性,种群的移动方向和步长均具有随机性,利用莱维飞行^[14]模拟该阶段,增强种群的探索性能。

$$\mathbf{X}_{i,:}^{t+1} = \mathbf{X}_{i,:}^t + \mathbf{S}_{i,:}^t \mathbf{K}_{i,:}^t (\mathbf{X}_{m,:}^t - \mathbf{X}_{i,:}^t) \quad (3)$$

式中, $\mathbf{X}_{m,:}^t$ 表示随机选择的个体, $\mathbf{S}_{i,:}^t$ 是莱维飞行限制因子由式(4)计算, $\mathbf{K}_{i,:}^t$ 是根据式(5)计算得出的自适应步长调节因子。

$$\mathbf{S}_{i,:}^t = 0.9 \times \tanh \left[4 \times \left(\frac{t_{gu}}{\alpha T_{max}} \right) \right] \quad (4)$$

式中, t_{gu} 是阵风阶段的迭代次数, α 是阵风阶段在整个迭代周期内的占比系数,文中取0.1。

$$\mathbf{K}_{i,:}^t = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (5)$$

式中, u 和 v 服从正态分布,具体形式见式(6)。

$$\begin{cases} u \sim N \left[0, \left\{ \frac{\Gamma(1+1.5) \sin\left(\frac{1.5\pi}{2}\right)}{\Gamma\left(\frac{1+1.5}{2}\right) 1.5^{2(1.5-1)/2}} \right\}^{2/1.5} \right] \\ v \sim N(0, 1) \end{cases} \quad (6)$$

3.1.2 种群探索行为

在初始觅食阶段中,种群已经确定了较高价值的搜索区域,这些信息储存在全局最优粒子中。该阶段的种群受全局最优个体影响运动,为了减少种

群多样性损失,引入随机粒子 \mathbf{X}_m 粒子位置信息,种群的探索行为可以描述为:

$$\mathbf{X}_{\text{ERA},i,j}^{t+1} = \mathbf{X}_{i,j}^t + \mathbf{R}_{i,j} \times (\mathbf{g}_{\text{best}} - \mathbf{X}_{z,i,j}) \quad (7)$$

式中, $\mathbf{R}_{i,j}$ 表示区间 $[0,1]$ 内的均匀分布随机变量, $\mathbf{X}_{z,i,j}$ 表示随机选择的种群中的非全局最优粒子。

3.1.3 种群开发行为

种群中的粒子受全局最优粒子和其他粒子的影响,个体的探索行为可以表示为:

$$\mathbf{X}_{\text{ETA},i,j}^{t+1} = \mathbf{X}_{i,j}^t + \omega' \varphi \sin(\varphi) |\mathbf{g}_{\text{best},j} - \mathbf{X}_{i,j}^t| + r(\mathbf{X}_{z,i,j}^t - \mathbf{X}_{i,j}^t) \quad (8)$$

式中, $r \sim U(0,1)$, $\varphi \sim U(0,2\pi)$, ω 是步长控制因子,步长控制因子随迭代次数 t 产生变化,由式(9)计算。

$$\omega' = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \times \frac{t}{T_{\text{max}}} \quad (9)$$

式中, ω' 是探索权重, ω_{max} 是权重的最大值, ω_{min} 是权重的最小值, T_{max} 是最大迭代次数。

3.2 种群的探索与开发的平衡模型

经过式(3)所示的初始觅食阶段后,剩余觅食过程种群动作由式(7)和式(9)生成,鸟群将受最优粒子吸引,向最优粒子聚集。在这个过程中,种群行为特征依次为开发、探索和开发。种群的位置更新方式:

$$\mathbf{X}_{i,j}^{t+1} = \begin{cases} \mathbf{X}_{\text{ERA},i,j}^{t+1}, & \text{if } r_w < p_w(t); \\ \mathbf{X}_{\text{ETA},i,j}^{t+1}, & \text{otherwise.} \end{cases} \quad (10)$$

式中, $r_w \sim U(0,1)$, $p_w(t)$ 由式(11)计算。

$$p_w(t) = p_w^{\text{end}} - \frac{t}{T_{\text{max}}} (p_w^{\text{end}} - p_w^{\text{init}}) \quad (11)$$

式中, p_w^{end} 和 p_w^{init} 分别是 p_w 的终值和初值,取值是 0.1 和 0.9, α_w 是该阶段在总迭代次数 T_{max} 中的占比系数, $t_w \in [1, \alpha_w T_{\text{max}}]$ 。

3.3 种群初始化及位置更新

假设种群规模是 N ,具有 D 个维度,种群的初始化方法如式(12)所示。

$$\mathbf{X}_{i,j} = lb + r_{\text{init}}(ub - lb) \quad (12)$$

迭代过程中粒子位置更新策略由式(13)计算,最优粒子位置由式(14)计算。

$$\mathbf{X}_{i,j}^{t+1} = \begin{cases} \mathbf{X}_{i,j}^{t+1}, & \text{if } F(\mathbf{X}_{i,j}^{t+1}) < F(\mathbf{X}_{i,j}^t); \\ \mathbf{X}_{i,j}^t, & \text{otherwise.} \end{cases} \quad (13)$$

式中, F 表示优化目标函数。

$$\mathbf{g}_{\text{best}}^t = \min(F(\mathbf{X}_i)) \quad (14)$$

在迭代过程中,根据迭代次数 t 选择对应的不

同的阶段策略,进而生成下一时刻的种群位置。

3.4 算法复杂度分析

IPSO 算法的计算复杂度主要取决于其适应度函数的计算复杂度。IPSO 算法初始化时,适应度函数计算约 N 次,复杂度记为 $O(N \times D)$ 。种群初始搜索阶段占算法迭代的 10%,计算复杂度可以表示为 $O(N \times D \times 0.1 T_{\text{max}})$; 剩余觅食阶段占算法迭代次数的 90%,计算复杂度是 $O(N \times D \times 0.9 T_{\text{max}})$; 整个算法过程的计算复杂度是 $O(N \times D + N \times D \times T_{\text{max}})$ 。PSO 算法的计算复杂度也为 $O(N \times D + N \times D \times T_{\text{max}})$,可见,IPSO 算法并未引入额外的增加计算复杂度的计算量。

3.5 基于 IPSO 算法的 PID 参数优化流程

图 1 直观地描述了应用于直流电机的基于 IPSO 算法的 PID 控制系统框图,IPSO 算法的作用是为 PID 控制器提供最优参数。

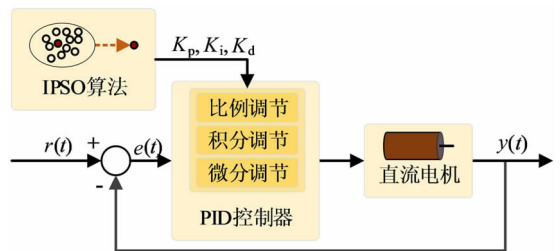


图 1 基于 IPSO-PID 的直流电机控制系统

适应度函数选用绝对误差积分指标(ITAE),具体形式为:

$$F(t) = \int_0^{\infty} t_{\text{pid}} |e(t_{\text{pid}})| dt \quad (15)$$

式中, t_{pid} 是受控系统模拟时间, $e(t_{\text{pid}})$ 是误差。

设计如图 2 所示的基于 IPSO 算法的 PID 参数优化流程,关键步骤可以概括为:

- (1) 根据参数初始化 K_p 、 K_i 和 K_d 的取值范围初始化种群,种群规模为 $N \times 3$;
- (2) 根据式(12)计算种群适应度;
- (3) 基于 IPSO 算法进行迭代寻优,在一次迭代 t 中完成 t_{pidmax} 的 PID 控制下的系统仿真;
- (4) 当迭代次数到达最大迭代次数 T_{max} 后输出最优参数 K_p^* 、 K_i^* 和 K_d^* 。

4 实验结果与分析

实验部分包括两个部分,第一部分在 CEC 2017^[16] 函数集上评估了 IPSO 算法性能;第二部分

评估了 IPSO 算法在 PID 控制器优化任务上的性能。所有实验在 Intel (R) Core i7-9700K CPU @ 3.50 GHz 的个人计算机上使用 Python3.8 进行。

4.1 IPSO 算法改进策略有效性评估

本节中对比了 IPSO 算法和 PSO 算法在 CEC 2017 函数问题上的收敛行为,旨在验证 IPSO 算法改进策略对算法性能的贡献。图 3 展示了种群中所有个体的平均适应度变化趋势,IPSO 算法在种群初始搜索阶段(迭代过程的前 10%)倾向于探索,平均适应度曲线下降较缓,这说明种群中的个体探索到不同适应度的区域;当迭代次数大于 10%后,种群在探索过程中兼顾开发,并随着迭代次数的增加进一步地倾向于开发,种群的平均适应度曲线加速收敛。从整体收敛曲线可知,相较于 PSO 算法,IPSO 算法收敛精度更高。

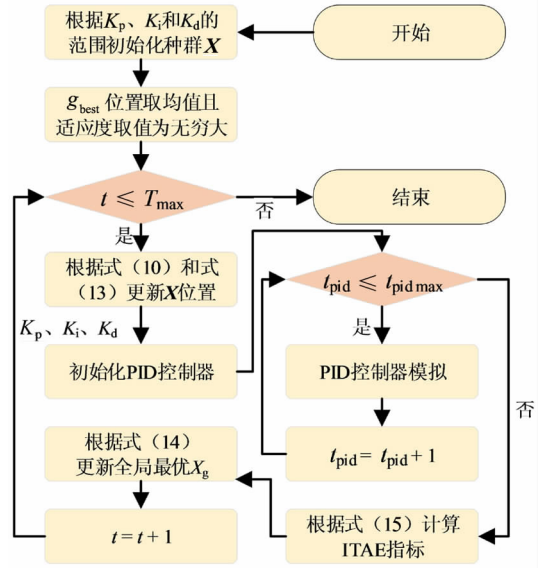


图 2 基于 IPSO 算法的 PID 参数优化流程图

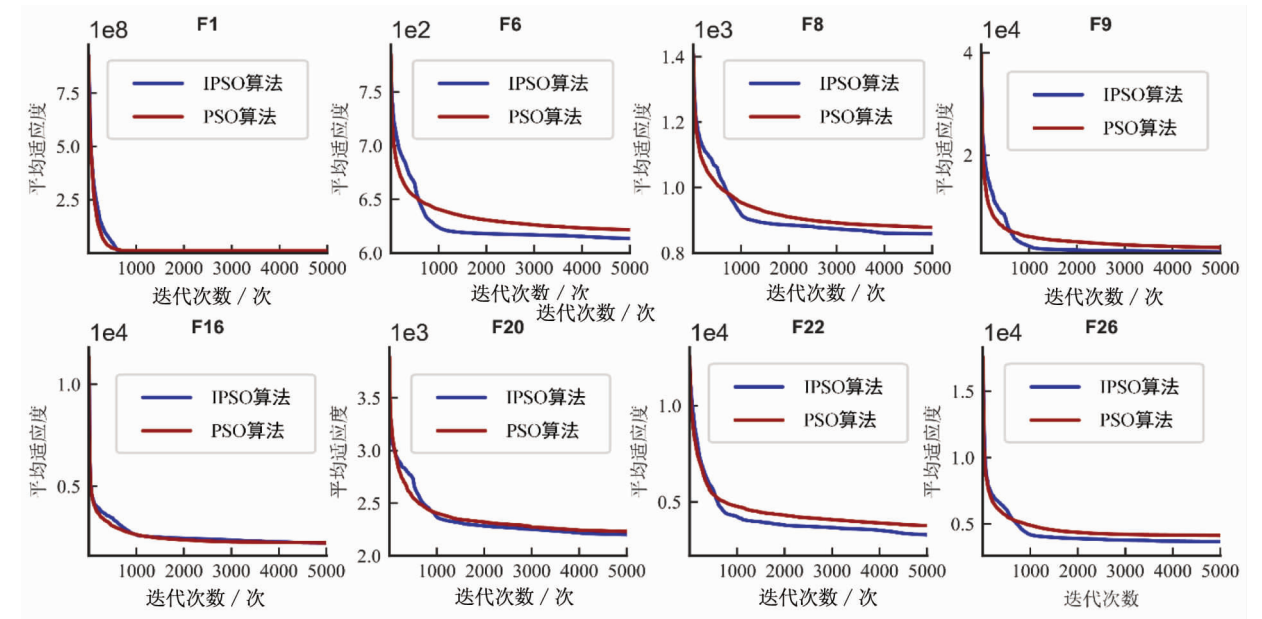


图 3 IPSO 算法与 PSO 算法收敛曲线对比

4.2 IPSO 算法数值分析

为了更全面地评估 IPSO 算法的求解性能,将 IPSO 算法与 PSO 算法、爬行动物优化算法 (Reptile Search Algorithm, RSA)^[17] 和自适应平衡优化算法 (Adaptive Balance Optimization Algorithm, ABOA)^[18] 进行对比。最大迭代次数设置为 5000,种群数量 N 设置为 30,每个函数独立运行 30 次,统计结果在表 1 中展示。

由表 1 中的统计结果可知,IPSO 算法为最优算法,ABOA 为次优算法,RSA 和 PSO 分别排名第 3 和第 4。其中,IPSO 算法的寻优结果中排名

第 1 的函数占总数的 82.76%,排名第 2 的函数占总数的 17.24%。实验结果表明,相较于对比算法,IPSO 算法具有更高的寻优精度,该结果证明了 IPSO 算法在寻优能力上更具竞争力。

4.3 仿真实验

直流电机被广泛用于机床、包装机和起重机等设备中,实验中的被控对象选取自参考文献[19]中的直流电机,PID 控制器的关键参数 K_p 、 K_i 和 K_d 取值上、下界分别为 60 和 0.1,种群规模 $N \times D = 20 \times 3$,最大迭代次数 $T_{max} = 100$ 次,受控对象的传递函数如式(16)所示。

$$W(s) = \frac{1.786}{0.0022s^2 + 0.17s + 1} \quad (16)$$

最优PID参数在表2中列出,此外表2中还列出了文献[20]的最优PID参数。图4展示了不同

优化算法在PID参数优化问题上的适应度曲线,从图中结果可知,IPSO算法收敛精度最高,最优适应度值为0.0184。

表1 不同算法在30维的CEC 2017问题上的实验结果

函数	指标	IPSO	ABOA	PSO	RSA	函数	指标	IPSO	ABOA	PSO	RSA
F1	均值	1.10E+02	1.00E+07	3.31E+06	1.48E+06	F16	均值	2.23E+03	2.24E+03	4.67E+03	3.37E+03
	排名	1	2	4	3		排名	1	2	4	3
F3	均值	8.65E+03	1.66E+04	1.67E+04	1.81E+04	F17	均值	1.87E+03	1.91E+03	3.22E+03	2.24E+03
	排名	1	2	4	3		排名	1	2	4	3
F4	均值	4.48E+02	4.48E+02	1.18E+03	1.01E+03	F18	均值	1.81E+03	1.86E+03	2.06E+04	1.62E+04
	排名	1	2	4	3		排名	1	2	4	3
F5	均值	5.56E+02	5.77E+02	8.94E+02	5.61E+02	F19	均值	1.94E+03	1.98E+03	1.08E+04	1.47E+04
	排名	1	3	4	2		排名	1	2	3	4
F6	均值	6.13E+02	6.22E+02	7.03E+02	7.40E+02	F20	均值	2.20E+03	2.23E+03	2.94E+03	2.53E+03
	排名	1	2	3	4		排名	1	2	4	3
F7	均值	7.89E+02	8.31E+02	1.39E+03	6.44E+02	F21	均值	2.35E+03	2.38E+03	2.68E+03	2.52E+03
	排名	2	3	4	1		排名	1	2	4	3
F8	均值	8.59E+02	8.78E+02	1.13E+03	9.90E+02	F22	均值	3.29E+03	3.77E+03	9.22E+03	3.60E+03
	排名	1	2	4	3		排名	1	3	4	2
F9	均值	1.11E+03	1.97E+03	1.15E+04	1.03E+03	F23	均值	2.71E+03	2.74E+03	3.53E+03	2.89E+03
	排名	2	3	4	1		排名	1	2	4	3
F10	均值	3.79E+03	4.07E+03	8.70E+03	3.57E+03	F24	均值	2.90E+03	2.94E+03	3.58E+03	3.05E+03
	排名	2	3	4	1		排名	1	2	4	3
F11	均值	1.15E+03	1.16E+03	1.28E+04	8.40E+03	F25	均值	2.89E+03	2.89E+03	4.68E+03	2.94E+03
	排名	1	2	4	3		排名	1.5	1.5	4	3
F12	均值	5.58E+03	8.49E+03	2.48E+04	1.50E+03	F26	均值	3.66E+03	4.14E+03	9.56E+03	5.40E+03
	排名	2	3	4	1		排名	1	2	4	3
F13	均值	1.40E+03	1.48E+03	2.77E+04	6.47E+03	F27	均值	3.21E+03	3.35E+03	3.83E+03	3.25E+03
	排名	1	2	4	3		排名	1	3	4	2
F14	均值	1.50E+03	1.51E+03	2.30E+03	3.06E+03	F28	均值	3.15E+03	3.18E+03	5.47E+03	3.30E+03
	排名	1	2	3	4		排名	1	2	4	3
F15	均值	1.54E+03	1.62E+03	7.71E+03	8.31E+03	F29	均值	3.50E+03	3.53E+03	6.09E+03	4.26E+03
	排名	1	2	3	4		排名	1	2	4	3
F30	均值	4.20E+03	4.29E+03	8.18E+04	2.74E+03	统计	算法	IPSO	ABOA	PSO	RSA
	排名	2	3	4	1		排序	1	2	4	3

表2 最优PID参数

参数	IPSO	PSO	RSA	ABOA	文献[20]
K_p	55.01	50.28	51.16	45.00	50.88
K_i	25.00	33.89	34.07	35.01	20.89
K_d	0.64	0.71	0.57	0.48	1.07

图5展示了基于表2参数获取的系统响应曲线,表3列出了对应的系统性能指标。结合表3和图5结果可知,基于IPSO算法获取的PID参数可以使控制器获取更优的控制性能。

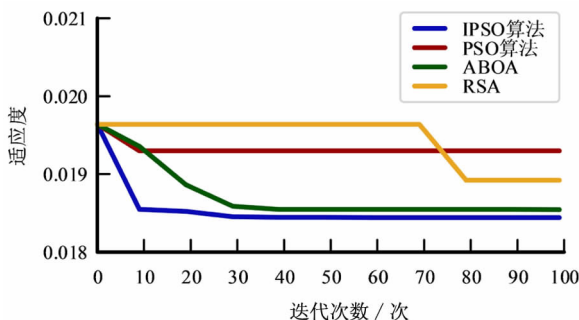


图4 PID参数优化任务的适应度曲线

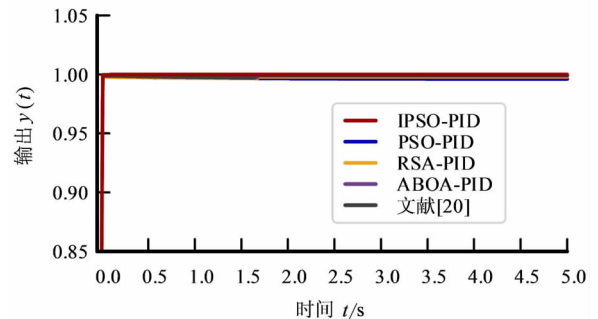


图5 系统阶跃响应曲线对比结果

表 3 不同参数场景下的性能指标

指标	IPSO	PSO	RSA	ABOA	文献[20]
超调量/%	0.00	0.00	0.00	0.00	0.00
上升时间/s	5.00E-05	5.50E-04	5.40E-05	6.10E-05	5.40E-05
调节时间/s	9.00E-05	9.90E-04	9.70E-05	1.11E-04	9.80E-05
稳态误差	0.000	0.015	0.002	0.001	0.001

5 结 论

针对 PSO 算法探索和利用不平衡问题导致降低算法寻优性能的问题,提出了 IPSO 算法。IPSO 算法考虑了不同阶段中探索和利用的平衡,建立了对应的位置更新模型,完善了 PSO 算法的寻优方法。将 IPSO 算法应用至 PID 参数优化任务中,实验结果表明,相较于对比算法,IPSO 算法可以使 PID 控制器获取更好的控制性能。

本文所提出的方法提升了 PSO 算法的寻优性能,下一步工作的重点是将 IPSO 算法应用至多关节机器人电机的 PID 控制器优化任务。

参 考 文 献

[1] GBADEGA P A, SUN Y X. Iterative LMI-tuned PID controller for robust decentralized multi-area automatic generation control in deregulated electricity markets[J]. International Journal of Engineering Research in Africa, 2024, 71: 109-140.

[2] OLEIWI B K, MOHAMED M J. Optimal design of linear and nonlinear PID controllers for speed control of an electric vehicle[J]. Journal of Intelligent Systems, 2024, 33(1): 1-8.

[3] LONG Z Y, LI J, YANG H W, et al. Methods and experiments for automatic control of surface back pressure based on dung beetle optimizer-PID controller[J]. Process Safety and Environmental Protection, 2024, 191: 360-374.

[4] HERNÁNDEZ-GUZMÁN V M, ORRANTE-SAKANASSI J. PID control of robot manipulators actuated by BLDC motors[J]. International Journal of Control, 2021, 94(2): 267-276.

[5] 吉建华, 苗长云, 李现国, 等. 基于 PSO 带式输送机 PID 控制器参数智能整定的适应度函数设计[J]. 机械工程学报, 2023, 59(22): 444-456.

[6] 刘尚波, 丹泽升, 廉保旺, 等. 干涉式闭环光纤陀螺仪的 PSO-PID 控制优化方法[J]. 红外与激光工程, 2024, 53(3): 250-261.

[7] 李欢欢, 刘辉, 盖江涛, 等. 基于粒子群优化算法 PID 参数

优化的双电机耦合驱动履带车辆转向控制[J]. 兵工学报, 2024, 45(3): 916-924.

[8] 朱凤磊, 张立新, 胡雪, 等. 基于蝙蝠优化 BP-PID 算法的精准施肥控制系统研究[J]. 农业机械学报, 2023, 54(S1): 135-143.

[9] 冯旭刚, 章义忠, 王正兵, 等. 基于萤火虫算法的 DDE-PID 加热炉炉温控制[J]. 中南大学学报(自然科学版), 2023, 54(9): 3688-3697.

[10] 吴沁, 周顺仟, 王星联. 改进粒子群优化滚珠丝杠进给系统 BP 神经网络 PID 控制策略研究[J]. 西安交通大学学报, 2024, 58(6): 24-33.

[11] 张稚荷, 周凯红, 朱梦岩. 基于改进 SSA 优化 PID 的下肢康复机器人控制[J]. 现代电子技术, 2024, 47(13): 153-159.

[12] 舒奕彬, 李立君, 张振翻, 等. 基于改进海马优化算法的 PID 参数优化[J]. 机床与液压, 2024, 52(13): 189-194.

[13] ZHANG Y, LI F, ZHANG Y, et al. Enhanced whale optimization algorithm for fuzzy proportional-integral-derivative control optimization in unmanned aerial vehicles[J]. Machines, 2024, 12(5): 295.

[14] HOUSSEIN E H, GAD A G, HUSSAIN K, et al. Major advances in particle swarm optimization: theory, analysis, and application[J]. Swarm and Evolutionary Computation, 2021, 63: 100868.

[15] SOMEFUN O A, AKINGBADE K, DAHUNSI F. The dilemma of PID tuning[J]. Annual Reviews in Control, 2021, 52: 65-74.

[16] WU G, MALLIPEDDI R, SUGANTHAN P. Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization[R]. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 2017: 1-16.

[17] ABUALIGAH L, ABD ELAZIZ M, SUMARI P, et al. Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer[J]. Expert Systems with Applications, 2022, 191: 116158.

[18] ZHANG C, LIU M, ZHONG P, et al. An adaptive balance optimization algorithm and its engineering application[J]. Advanced Engineering Informatics, 2023, 55: 101908.

[19] 郑修斌, 陈珺, 刘飞. 改进哈里斯鹰算法的 PID 自整定及在直流电机调速中的应用[J]. 控制工程, 2023: 1-10.