

[PID参数优化]

基于混沌自适应果蝇优化的PID参数优化

文/冷欣 周腾飞

东北林业大学计算机与控制工程学院

针对标准果蝇算法 (FOA) 收敛速度快慢依赖初始化且易陷入局部最优的缺陷, 提出一种混沌自适应果蝇优化算法 (GIFOA) 的PID参数优化方案。GIFOA采用混沌映射初始化种群, 增加种群的遍历性; 自适应调节搜索步长平衡全局和局部搜索能力; 以搜索方向作为浓度判定值, 控制偏差绝对值和控制器输入平方项的时间积分作为适应度函数, 并将此优化算法应用于二阶时滞系统的PID控制参数调节, 并通过迭代寻优得到最优控制器参数。结果表明, GIFOA适应度值收敛速度快、准确度高; 阶跃响应的控制准确度高, 响应速度更快, 鲁棒性好, 优于标准的果蝇优化算法, 为PID参数优化提供了参考。

PID控制器是一种经典的控制算法, 由于其原理简单、易于实现和稳定性好等特点, 被广泛应用于各行各业, 如工业控制和过程控制中流量、温度、压力和速度等^[1]。PID控制效果好坏取决于参数选取, 常见的参数调整方式有临界比例度法^[2]、衰减曲线法^[3]和经验法等, 但手动调参的方法费时费力, 依赖调节人员的经验, 且难以得到理想的最优参数。近年来, 随着智能技术的发展, 有学者提出采用智能优化算法调整PID控制器的参数, 如粒子群算法^[4]、遗传算法^[5]、鲸鱼算法^[6]和麻雀算法^[7]等算法, 并取得一定效果。这类智能优化算法简单易理解, 在解决控制器参数调整方面有广阔的应用前景。然而, 遗传算法编程实现复杂, 粒子群收敛速度慢且易陷入局部最优解。

果蝇优化算法 (Fruit-fly Optimization Algorithm, FOA) 是台湾学者潘文超提出的一种新的全局优化算法^[8]。该算法是一种基于果蝇觅食行为而推演出的全局优化算法, 应用于各行各业如PID参数优化^[9]、求解函数优化和支持向量机^[10]等。FOA具有原理简单、代码易实现、收敛速度快和参

数少等优点。LIU等采用基于混沌搜索的FOA算法整定PID参数, 减少了适应度波动^[11]; 李明辉等人采用自适应步长的FOA算法整定PID参数, 提高了收敛速度^[12]。

本文针对标准果蝇算法 (FOA) 收敛准确度、易陷入局部最优解的缺陷, 提出一种混沌自适应果蝇优化算法 (GIFOA) 的PID参数优化方案。通过利用混沌特性初始化来提高果蝇种群的多样性和搜索的遍历性; 同时通过引入自适应半径调节搜索步长, 提高算法收敛速度和收敛准确度, 避免陷入局部最优。结果表明, 该优化算法可快速高效地找到最优参数, 提高PID控制的性能参数。

PID基本原理

PID控制, 也称为比例 (Proportional) 积分 (Integral) 微分 (Derivative) 控制, 能够将系统误差线性组合后反馈到被控对象上。由于其原理简单、易实现和参数少等优点, 广泛应用于各行各业。其工作原理为: 首先求出期望值与实际输出间的误差值, 然后将之送入比例、积分和微分三个模

块中,输出的组合量即为闭环控制量, 将之送入被控对象以使输出不断向期望值靠近, 最终实现对被控对象的控制。PID控制的闭环系统由PID控制器和被控对象构成, 如图 1 所示。

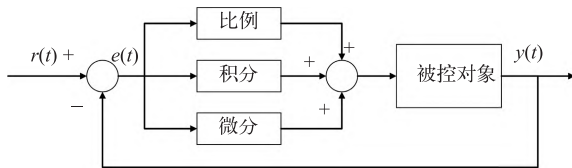


图1 PID控制框图

给定值与实际值之间的偏差值 $e(t)$ 为

$$e(t) = r(t) - y(t) \quad (1)$$

式中, $r(t)$ 为给定期望值; $y(t)$ 为系统输出值。

PID传递函数表达式为

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (2)$$

式中, K_p 为比例系数; $K_i = K_p/T_i$ 为积分增益; $K_d = K_p/T_d$ 为微分增益; T_i 为积分时间常数; T_d 为微分时间常数。

果蝇算法及其改进

1. 标准果蝇优化算法

果蝇的嗅觉和视觉非常敏感, 通过自身对气味的敏感度, 捕捉空气中各种气味, 快速寻找远方的食物源; 飞到食物位置附近, 以敏感的视觉特性为基础, 找到食物源和其他果蝇的位置, 并向该方向移动。果蝇与食物源的距离与味道浓度相关, 距离越近, 味道浓度越高^[13], 整个搜索食物的过程即从味道浓度低的位置移向味道浓度高的位置。果蝇群体觅食过程如图2所示。

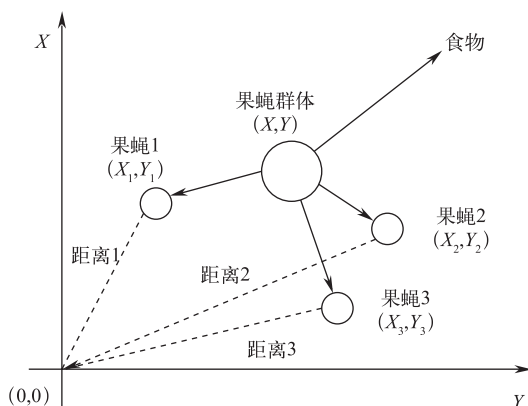


图2 果蝇群体觅食过程

果蝇优化算法具体实现步骤如下。

1) 设置算法种群规模、迭代次数和维度等参数, 初始化果蝇群体位置 X_axis 和 Y_axis 。

2) 给出果蝇随机飞行方向和距离为

$$\begin{aligned} X_i &= X_axis + 2rand - 1 \\ Y_i &= Y_axis + 2rand - 1 \end{aligned} \quad (3)$$

式中, $rand$ 为(0,1)随机值。

3) 计算果蝇个体 i 与原点之间的距离 D_i , 再利用距离的倒数作味道浓度判定值 S_i 为

$$\begin{aligned} D_i &= \sqrt{X_i^2 + Y_i^2} \\ S_i &= 1/D_i \end{aligned} \quad (4)$$

4) 将判定值 S_i 带入味道浓度函数(目标函数) $F(S_i)$ 中, 计算果蝇个体 i 的味道浓度值为

$$Smell_i = F(S_i) \quad (5)$$

5) 找到果蝇群体中最优浓度值和其果蝇个体为

$$[bestSmell, bestIndex] = \min(Smell) \quad (6)$$

6) 存储最优浓度值和坐标记录, 群体果蝇飞向该位置为

$$\begin{aligned} [Smellbest] &= bestSmell \\ X_{axis} &= X(bestIndex) \\ Y_{axis} &= Y(bestIndex) \end{aligned} \quad (7)$$

7) 进入迭代寻优, 重复执行步骤2)~步骤5), 判断当前味道浓度是否高于前一次最佳浓度值, 且当前迭代次数是否小于最大迭代次数, 若是则记录下来

2. 改进果蝇算法

(1) 搜索步长改进

基本FOA中第2)步搜寻范围在 $[-1, 1]$ 之间, 即搜索半径为1的区间。这会导致前期因搜索步长过小导致搜索效率低下, 影响收敛速度和全局寻优能力, 在搜索后期又因搜索步长过大而错过全局最优解, 影响局部寻优能力和寻优准确度。为此引入自适应半径调节系数, 在搜索前期可增大搜索步长, 增加全局寻优能力, 提高收敛速度; 在搜索后期减小搜索步长, 增强局部寻优能力, 提高收敛准确度^[14]。改进后果蝇个体随机飞行方向和距离为

$$X_i = X_axis + w(2rand - 1) \quad (8)$$

$$w = w_{max} \exp \left(\log \left(\frac{w_{min}}{w_{max}} \right) \times \frac{Iter}{Iter_{max}} \right) \quad (9)$$

式中, w_{max} 为最大搜索半径; w_{min} 为最小搜索半径;

Iter为迭代次数；Iter_{max}为最大迭代次数。

在迭代初期，根据较大的搜索步长进行全局搜索，在搜索后期随着迭代次数增加，搜索半径减小，进行局部搜索。

(2) 搜索方向改进

标准果蝇算法在第3)步计算果蝇味道浓度判定值，其值始终为正，从而得出适应度函数不精确。本文令气味浓度判定值 $S_i=X_i^{[15]}$ ，从而计算最小气味浓度值。

(3) 混沌初始化改进

混沌映射^[16]是一种利用混沌运动特性进行优化搜索方法，其中Logistic映射具有设计简单、编程易实现、运算效率高等优点，故采用Logistic映射为

$$x_{k+1} = \alpha x_k (1 - x_k) \quad (10)$$

式中， k 为迭代次数； $x_k \in (0,1)$ ； α 为控制参数。

当 $\alpha \in (3.56, 4]$ ，式(10)产生混沌系统进入混沌状态，当 α 越接近4时，混沌状态越强烈。

(4) 混沌映射具体步骤

1) 执行标准果蝇优化算法第1)步，得到 X_{axis} 。

2) 将果蝇初始位置 X_{axis} 映射为混沌变量 $CX_{axis}^{[17]}$ ，初始位置位于 $[a, b]$ 内， a 、 b 可根据实际情况选取，混沌变量取值范围为 $(0, 1)$ ，即将 X_{axis} 从 $[a, b]$ 映射到 $(0, 1)$ 上的公式为

$$CX_{axis} = \frac{X_{axis} - a}{b - a} \quad (11)$$

3) 将混沌变量 CX_{axis} 通过公式(10)做混沌操作，得到变量 CX 为

$$CX = \alpha CX_{axis} (1 - CX_{axis}) \quad (12)$$

4) 将混沌变量 CX 再次映射为普通变量 X'_{axis} ，即将从区间 $(0, 1)$ 映射到区间 $[a, b]$ 上的公式为

$$X'_{axis} = a + CX(b - a) \quad (13)$$

基于混沌自适应果蝇算法的PID控制器设计

1. 适应度函数设计

GIFOA搜索过程中，采用气味浓度值 S_{meil} 作为更新评定值，气味浓度函数 $F(S_i)$ 是智能优化算法GIFOA和PID控制联系的桥梁，因此设计合适的适应度函数（又称目标函数或气味浓度函数）至关重要。

为防止误差过大而影响控制性能，采用误差绝对值的时间积分性能指标作为目标函数其中一项。为进一步避免控制量过大，加入控制器的输出平方项在内，使得目标函数由误差量 e 和控制器输出项 u 同时控制^[18]。则目标函数为

$$J = \int_0^\infty [w_1 |e(t)| + w_2 u^2(t)] \quad (14)$$

式中， $e(t)$ 为输入量 $r(t)$ 和系统输出 $y(t)$ 之间的误差值； $u(t)$ 为控制器输出； w_1 和 w_2 为权值常数。

为了避免超调，采用一定的惩罚机制，将超调量作为最优指标的一项，此时目标函数为

$$J = \begin{cases} \int_0^\infty [w_1 |e(t)| + w_2 u^2(t) + w_3 |ey(t)|] dt & |ey(t)| < 0 \\ \int_0^\infty [w_1 |e(t)| + w_2 u^2(t)] dt & |ey(t)| > 0 \end{cases} \quad (15)$$

$$eyt = y(t) - y(t-1) \quad (16)$$

式中， w_3 为权重常数，且 $w_3 \gg w_1$ ，一般情况下， $w_1=0.999$ ， $w_2=0.001$ ， $w_3=100$ 。

2. PID控制器的GIFOA算法设计流程图和实现过程参数优化框图如图3所示。

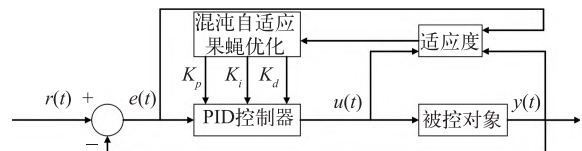


图3 基于GIFOA的PID参数优化框图

3. GIFOA流程图如图4所示。

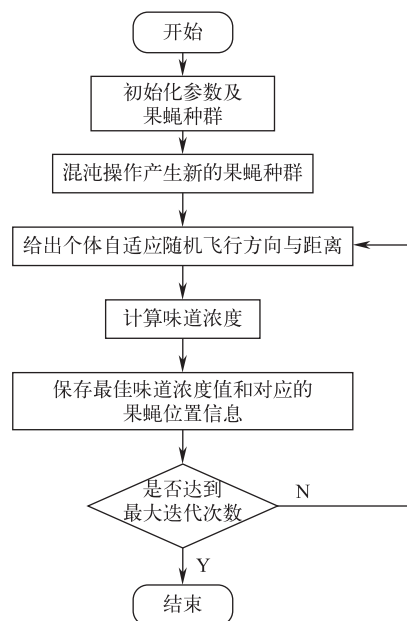


图4 混沌自适应果蝇优化算法流程图

GIFOA实现步骤如下:

1) 设置算法种群规模、迭代次数和维度等参数, 初始果蝇群体位置 X_{axis} , 其中每个果蝇位置由 K_p 、 K_i 、 K_d 组成

$$X_{axis} = [K_p K_i K_d] \quad (17)$$

2) 执行混沌映射步骤2)~步骤4), 得到混沌初始化种群 X'_{axis} 。

3) 给出GIFOA果蝇随机飞行方向和距离

$$X_i = X'_{axis} + w(2rand - 1) \quad (18)$$

4) 令气味浓度判定值 $S_i = X_i$, 根据式(15)和式(16)计算出气味浓度的适应度值为

$$Smell_i = F(S_i) \quad (19)$$

5) 执行FOA步骤5)~步骤7)。

仿真实验与结果分析

1. 被控对象选取和仿真参数确定

(1) 被控对象选取

工业控制中, 许多系统近似为一阶或二阶的典型系统, 本文选用二阶延迟系统作为被控对象, 分别用FOA和GIFOA对该系统PID控制器参数进行整定。二阶延迟系统为

$$G(s) = \frac{400}{1s^2 + 1s} e^{-2s} \quad (20)$$

(2) 仿真参数确定

设果蝇种群大小为40, 最大迭代次数为100, 控制参数 α 为3.9, 最大搜索半径 w_{max} 为5, 最小搜索半径 w_{min} 为0.01, 空间维度为3, 初始化种群 X_{axis} 的初始化范围下界 a 为0.01, 上界 b 为20。 K_p 范围为[0,100], K_i 范围为[0,20], K_d 范围为[0,20], 给定输入为阶跃信号 $r(t)=1$, 仿真步长为0.001 s。

2. 仿真结果及分析

对该系统采用FOA、GIFOA两种算法进行仿真, FOA和GIFOA算法优化个体气味浓度适应度函数曲线如图5a所示, 图5b为系统阶跃响应输出曲线, 两种算法得到的PID优化参数见表1。

由图5a可看出: GIFOA比FOA算法准确度高、收敛速度更快, 初始适应度值更小, 更容易得到全局最优解。从图5b和表1可看出: 改进后的GIFOA优化PID参数上升时间比FOA快0.1 s, 调节时间比FOA快0.024 s, 且无超调。因此, GIFOA优化PID参数的方法比FOA控制准确度高、收敛速度快, 说

明GIFOA算法的有效性和优越性。

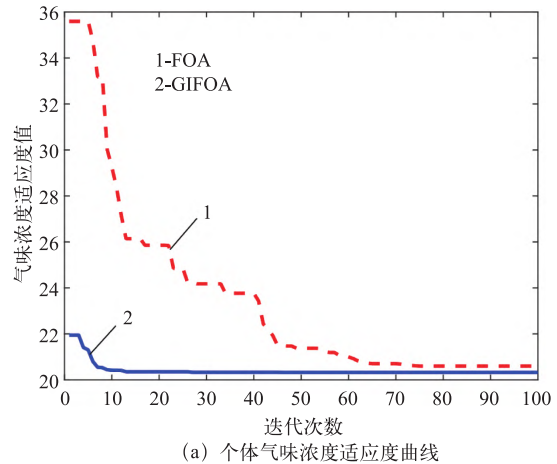
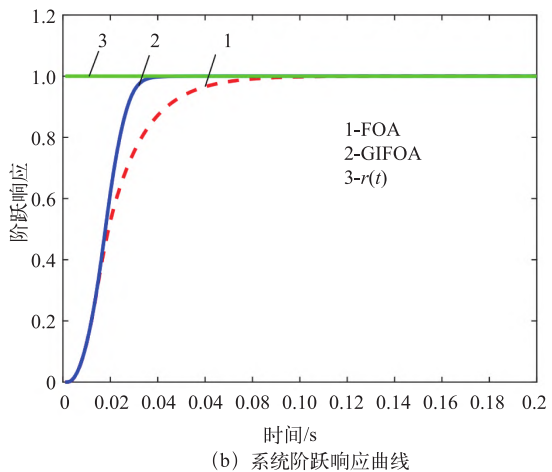


图5 气味浓度适应度和系统阶跃响应图

表1 FOA和GIFOA优化PID参数的性能指标

参数	GIFOA	FOA
K_p	52.066 3	40.52 09
K_i	0.5695	0.498 9
K_d	0.922 0	0.945 8
上升时间/s	0.23	0.33
调节时间/s	0.031	0.055
超调量 (%)	0	0
迭代次数/次	30	75
适应度	20.3	20.6

结束语

对于PID手动调节参数难以得到最优解的不足, 提出一种GIFOA的算法整定PID控制参数。通过二阶时滞系统仿真, 验证该算法的有效性和优越性。与FOA算法比较可知, GIFOA初始适应度值

更小,说明了混沌映射有效增加种群初始化的遍历性;GIFOA收敛速度更快和收敛准确度更高,可以有效避免陷入局部解,说明了自适应调整搜索步长的优势;调节时间和上升时间更快。因此,基于FIFOA算法的PID参数优化对工业控制具有一定的参考价值。**EA**

(收稿日期:2024-01-26)

参考文献

- [1] 何佳佳,侯再恩.PID参数优化算法[J].化工自动化及仪表,2010,37(11):1-4.
- [2] 于明星,王瑛,邵帅.基于Matlab的临界比例度法在工程PID参数自整定数值模拟中的应用[J].辽宁师专学报(自然科学版),2018,20(2):6-8+14.
- [3] 刘霞,冯元,王亚云.基于衰减曲线法整定PID调节器参数仿真技术研究[J].国外电子测量技术,2019,38(8):101-104.
- [4] 孙超,郭乃宇,严明蝶,等.改进自适应粒子群算法的PID参数优化[J].中国工程机械学报,2023,21(5):377-382.
- [5] 夏知胜,陈诚,杨爱斌.基于遗传算法优化PID的固定翼无人机俯仰控制设计[J].科技创新与应用,2023,13(34):41-44.
- [6] 王勇,肖平.基于改进鲸鱼算法的PID参数整定系统设计[J].芜湖职业技术学院学报,2023,25(1):43-46.
- [7] 张琴,蔡慧茹,兰明东,等.基于改进麻雀优化PID的波浪补偿控制方法[J/OL].工程科学与技术:1-15[2024-01-24].
- [8] PAN W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012(26): 69-74.
- [9] 宋娟,李萍.基于免疫果蝇优化算法的PID控制参数优化[J].控制工程,2017,24(12):2502-2507.
- [10] 郭凡,丁永生,郝矿荣,等.基于果蝇算法优化支持向量回归机的纺丝性能预测[J].系统仿真学报,2014,26(10):2360-2364.
- [11] LIU Y, WANG X, LI Y. A modified fruit-fly optimization algorithm aided PID controller designing[C]//Proceedings of the 10th World Congress on Intelligent Control and Automation. IEEE, 2012: 233-238.
- [12] 李明辉,曹泽,王玉洁.基于自适应果蝇优化算法的PID参数优化[J].机床与液压,2018,46(20):144-147.
- [13] 郭晓东,王丽芳,张学良.基于自适应步长的果蝇优化算法[J].中北大学学报(自然科学版),2016,37(6):570-575.
- [14] PAN, QUAN-KE, et al. An improved fruit fly optimization algorithm for continuous function optimization problems[J]. Knowledge-Based Systems, 2014(62): 69-83.
- [15] SHAN D, CAO G H, DONG H J. LGMS-FOA: an improved fruit fly optimization algorithm for solving optimization problems[J]. Mathematical problems in engineering, 2013(10): 1256-1271.
- [16] VAIDYANATHAN S. Chaos in neurons and adaptive control of Birkhoff-Shaw strange chaotic attractor[J]. International Journal of PharmTech Research, 2015, 8(5): 956-963.
- [17] 廖建庆,王涵,王咸鹏.混沌动态步长果蝇优化算法[J].传感器与微系统,2019,38(8):139-142.
- [18] 余胜威,曹中清.基于人群搜索算法的PID控制器参数优化[J].计算机仿真,2014,31(9):347-350,373.